

Annotated Help 3 Programs

vol. II

edit	20	pages
exit	2	-
move	8	-
print, pair	20	-
run	2	-
slip	25	-
start	2	-
system punch	11	-
check bin	2	-
cattap	1	-
punch head bin 0	1	-
punch head kompud	4	-
create new -> old	1	-

December 1967

REGNE CENTRALEN

[18.7.67] [edit/reservation of buffer areas] [page 1]
[character table]

[STOP, CLEAR]

b k=dd1, i=90, b60, e60 ; edit, outermost block

<d41, e1=-166, x e1=10 ; input buffer in Pass 1-2
> ; used by line buffer, reversetracks,
; INIT MEDIUM in Pass 1
<d41, e2=-86 , x e2=-86 ; byte input buffer in Pass 2
> ; INIT MMEDIUM inn Pass 1
<d41, e3=10 , x e3=-166 ; output buffer in Pass 1-2
>

[marks in e3-79e3 are initialized at start pass 1 as follows:

39e3 A mark
79e3 C marks, all others are zeroes]

[indicator usage: in Pass 1		in Pass 2
LZA	N in2nd	freely used
LZB	L endOK	-
LTA	not used	-
LTB	-	N case free
LPA	A mark of tableword	N only input sum
LPB	B -	N copy
LQA	N underlined	X external output
LQB	N sum	freely used
LRA	not used	blind
LRB	X typewriter input	N ready

where L,N or X indicates initialization. Used compound conditions:
NZC, LPC NZC, LPC, NRC

e4:

b a5

[character table:

The i-th entry in the following table describes the type of the character with value i. The entries can be freely changed by appropriate parameters. (cf. table change, page 12) Changing the description for LC, UC, TF or <10> can, however, interfere with the internal logic.

	description	parameter for edit
normal	qqf	1 <chv>, n
alarm	qq -2,	1 <chv>, a
blind	hvf s1 ,	1 <chv>, b
skip	qq 512	1 <chv>, s
end	qq 770	1 <chv>, e
case indep.	qq 896	1 <chv>, c
replaced	qq <chv>	1 <chv>, r, <chv>

where <chv> ::= <help number>

The descriptions for LC, UC, <10>, CLEAR and SUM are initialized to execute special actions, not available for assignment to other characters]

a3: pmf 0 , zqn(r+i+1) ; constant for sum = true
[1]

e5:

```

[0]   hvf s1,
[1]   qqf
[2]   qqf
[3]   qqf
[4]   qqf
[5]   qqf
[6]   qqf
[7]   qqf
[8]   qqf
[9]   qqf
[10]  qq   832           ; TEN: goto trailing edge;
[11]  qq   770
[12]  hvf s1,
[13]  qq   11
[14]  qqf
[15]  qq   -2,
[16]  qqf
[17]  qqf
[18]  qqf
[19]  qqf
[20]  qqf
[21]  qqf
[22]  qqf
[23]  qqf
[24]  qqf
[25]  qqf
[26]  qq   -2,
[27]  qqf
[28]  paf a48, hhf a1    ; CLEAR: goto clearcode;
[29]  hvf s1,
[30]  hvf s1,
[31]  hvf s1,
[32]  qqf
[33]  qqf
[34]  qqf
[35]  qqf
[36]  qqf
[37]  qqf
[38]  qqf
[39]  qqf
[40]  qqf
[41]  qqf
[42]  qq   -2,
[43]  qqf
[44]  hvf s1,
[45]  qq   -2,
[46]  qq   -2,
[47]  qq   -2,
[48]  qqf
[49]  qqf
[50]  qqf
[51]  qqf
[52]  qqf
[53]  qqf
[54]  qqf
[55]  qqf

```

```

[56] qqf
[57] qqf
[58] paf b , hvf s1 ; LC: case:= 0; blind action;
[59] qqf
[60] hvf a , ; UC: case:= 128; blind action;
[61] pmf s3 , hvf a2 ; SUM: goto sumcode;
[62] hvf s1 ,
[63] qq 512
[64] hvf s1 ,

```

[The following code up to 127e5 is treated as the continuation of the
table therefore it must not contain any f marked word]

```

b: qq [case] ;
a2: grn s3 MA ; sumcode: save slow action:= slow action;
gm b26 , hv s1 ; slowaction:= check;
a: pa b t 128 ; return;
a1: hv s1 , pa b1 ; clearcode: insum:= 0; sum:= true;
hv a4 LPC ; if first pass then blind action;
pm a3 , gm b2 ; insum2:= 0; A:= true;
e45: hv a4 NQA ; set summing: if -, external then begin
pa e8 Vt e25 NRB ; if -, ready V caution then out:= outinternal and check
e47: pa e8 t e26 IPA ; else begin out:= outinternal; only input sum:= A;
a4: hv e35 M ; end end; blind action;

```

e
[Start pass 2, stack updating:

The string to be searched for is input from the drum and placed from
e10 onwards. The string is terminated with the byte -512. Byte -511 in
the input stream means the end of the corrections.]

b a30

```

e6: a: qq[count], pp 0 ; START PASS 2: nextstring: i:= 1;
a1: pm (b15) Xt 1 ; BYTE:
hs e33 NA ; stack[i]:= Raddr:= char from drum;
gr pe10 M ; marks[i]:= 0;
pi (pe10) Vt -130 LT ; if Raddr > 0 then begin i:= i + 1; goto BYTE end;
pp p1 , hv a1 ; ready:= Raddr = -511 ; case free:= false;
pm r LPB ; copy:= -, copy;
bs p IPB ; if i > 0 then begin
acn pe9 V MA ; set A mark(stack[i+1]); skip line end;
hvn a11 NRB ; if -, ready then goto STRING FOUND;

```

```

a2: pp 1      V      ; BEGIN SEARCH: p:= 1; skip line;
a3: hs (a4)    LRA ; PERHAPS BLIND: if blind then execute(action[char]);
                        ; blind action:

```

[The next two instructions are changed if the input unit is the tape reader or the typewriter to:

```

e7: lyn e18    V      ; Raddr:= lyn; skip line;
[1e7]hv e13    , xx    ; RESET AND START PRESSED: go to trailing edge;]

e7: [-3] pm e18 Xt 1    ; NEXT CHAR:
[1e7]hv e22      LA      ; Raddr:= input;
[-1] ga a4      V      NT ; if parity V slowaction  $\neq$  0 then
    [changed to qq 0 if print or to qq 0, if check]
[3e7]hs e36      ; treat slow;
a4:b2: pm 0      Vt e5   IRA ; TEST: char:= Raddr + table base;
    [char] [sum]      ; M:= set marks(cell[char]); blind:= Amark;
b1: ac [insum]Dt 1      ; if sum then insum:= insum + Raddr + 1;
a5: hh a13      X      NB ; AFTER CENTRAL: if spec then go to TREAT SPECIAL;
e8: hs e26[out] LPB ; OUTPUT: if copy then out (Raddr);
    pm (a4)      XVD     NRC ; if blind V ready then go to PERHAPS BLIND;
    hv a3        ; Raddr:= char;
    ar b         NTA ; if -, case free then R:= char 1:= Raddr + case;
    ga a8        ITA ; case free:= false;

```

[char contains now the next character to be compared. Note that the characters in the stack are provided with their case (LC for case independent) and are also increased with e5 (cf. a4)]

```

sr pe9      , pp p1    ; R:= R - set marks(stack[p]); p:= p + 1;
hv a10      LZ      ; if R = 0 then goto TEST MARK;

```

[The comparison has failed. The following code decides whether stack[1:p-1] = stack[i:p-2] concat char, where i = 2, 3, ... p-1]

```

a9: pm p      DX      ; for s:= 2 step 1 until p do begin
    ps 1      , ps s1  ;   for i:= s step 1 until p do
    ca s      , hv a2   ;   if (if i = p then char else stack[i])
    pa a7     t e9      ;    $\neq$  stack[i-s+1] then goto SHORTEN;
a6: pp s+1    , pp p1   ; NEW CHANCE:
a7: pm e9     t 1       ; comment the last p-s+1 characters read
    ca p1     X         ; match with stack[1:p-s+1];
a8: sr [char]DV ;
    sr pe9    V         ;
    pp (a7)   Vt e11 LZ [e11=-e9+1]; p:= p - s + 2; goto NEXT CHAR;
    hh a6     X         LZ ; SHORTEN: end;
    hh a9     X         NZ ; go to BEGIN SEARCH;
                        ; TEST MARK:
a10: hv e7     NA      ; if NA then go to NEXT CHAR;
                        ; STRING FOUND: comment R = 0;

```

```

a11: can(b)          NPB ; out(if -; copy V case = 0 then
e12: am 58          DV   LZ ; 58 else 60);
      [used in pass 1 1a23] ; if copy then
      pm 60          DX   ; begin count:= count + 1; go to nextstring end;
a12: hs (e8)         ; IN SECOND STRING:
      pm (b15)       XVt 1 LPB ; Raddr:= char from drum;
      hh (a)          Dt 1   ; if Raddr > 0 then OUTR:
      hs e33          NA   ; begin out(Raddr);
      hv a12          NT   ; go to IN SECOND STRING end;

```

[Clearcode appears as -28. -512 marks the end of the string]

```

nc -28 , hh a ; if Raddr ≠ -28 then go to nextstring;
ps a12-1 , mt b11 ; alter(blindaction) to proceed to: (OUT R);
a13: hv e45 , ga a14 ; Raddr:= 28; go to set summing;
      bs (a4) t 64e5 NRA ; TREAT SPECIAL: swap; if char > 64e5 V IRA then
      ps 3e7 , hv e43 ; begin set return to(NEXT CHAR);
      sc (b1) DVt -1 NT ; treat char end; if R > 0 then begin
a14: pi 0 Vt 573 IZA ; replaced: insum:= insum - Raddr - 1;
      ga b2 , hv a4 ; char:= Raddr; go to TEST end else
      pm (e7) X NZB ; in:= Raddr bits 1 to 3; blind:= IZA:= false;
b48: ac[insum2] DVt 1 NT ; if NZB then skip: begin insum2:= insum2 +
      hv e8 X LTA ; last input char + 1; go to NEXT CHAR end;
      hv e7 NT ; if case free then go to OUTPUT;

      hv e13 LTB ; if ten then go to trailing edge;
      hv e8 X NRB ; ENDMARK: if -, ready then begin blind:= true;
a15: pa b38 Xt 10.3 ; go to OUTPUT end; wordend:= 10;
      ps r1 , hv (e8) ; out(Raddr); i:= 239; skip line;
e13: pa a16 , ud a15 ; trailing edge: wordend:= 10; i:= 240;
[2] hv a18 NQA ; if -, external then begin
a16: it -1 , qq (a17) ; for i:= 1 step -1 until 1 do
[-2] pm 10 DX ;
a17: bt 240 t -1 IZA ; outinternal(10); IZA:= false;
      ps r-3 , hv e26 ;
<d41, ;
e15: pp -39 , ps r ; for i:= 1 step 1 until filler do
[s1] ncn p ; next word out;
      pp p-1 , hv e28 ;
>a18:vy d43 , pa a20 ; select(alarm unit); print:= false; skip line;
e16: ; alarmprint: print:= true;
b10: vk 960[=+1.3 see 3e20], it -1 ;
      vk 1d37+d9, it -40 ; fetch last tracks of HELP system;
      lk -6 , it -1 ;
      bt 4 , hh e16 ;
a20: bs 1[print], hv c24 ; if print then go to ALARM PRINT;
<d41,ncn(e15) , hs a23 ; if filler ≠ 0 then begin only sense:= true; SENSE;
e44: can 1 , hv a22 > ; end; if tape output then go to ADJUST TAPE;

```

[Termination of the program:]

```

a24: am c1 , hs c2 ; CALL HELP: fetch catalog;
b5: pi [kind] , am a ; in:= kind; if count ≠ 0 then
      nc 0 , hs e39 ; display(count);
<d41,am b51 , ck 10 ; R:= if filler ≠ 0 then bits(10,21,paramword)
      tk -28 , ud c24 ; else outtrack;
      ca (e15) , am b23 ; prepare help entry to set twr;
x am b23 , ud c24 ; M:= first out block;
> pm b25 , hv c74 ; go to ADJUST SPECIAL;

```

```

<d41,
b6: gr i      V      ; see e27
[1b6]qq b6.9+1.19+2574.39 ; buffer base
[2b6]qq b6.9+1.19+2574.39 ; current base
[3b6]qq -d53.9-1.21-2574.39+d53.39 ; form checkword
[4b6]qq 1024.39 ; buffer length

```

```

b7: qqf15.5+15.11+15.17+15.23+15.29+15.35+3.39, ; filemark
[1b7]qq b7.9+1.19 ; filemark to buffer
[2b7]qq 1.19 ; filemark to tape
[3b7]qq b19 t 1 ; status to buffer, to core

```

[Adjust tape: cf. page 13 and Help 3 page 13]

```

a22: vk d11 , lk c ; ADJUST TAPE: select parameter track;
      vk (c5) , vk (c4) ; fetch search;
      lk d14 , vk (c4) ; fetch selected catalog track;
      pm b23 , tl 23 ;
      arm(c15) , ck 8 ; store[iword]:= store[iword] ^
      tl 16 , ck 16 ; 8m16016m V (block count pos 23);
      gr (c15) , hs c8 ; sum track;
      sk d14 , arm 1b7 ; write catalog track;
      us , arm 2b7 ; paramword:= filemark to tape;
      gr b52 , us (b35) ; write filemark;
      pa b49 , ps a24-1 ; up:= true; only sense:= true;
a23: pa b55 , hv e46 ; SENSE; go to CALL HELP;

```

>

e

[Procedure outchar outputs one character to an external unit. The output format is defined as follow:

- i. The case for the character SPACE or CARRET is always equal to the case of the previous character. (to undefined case in the beginning of the tape)
- ii. The procedure does not output superfluous case shifts.
- iii. If the first character (disregarding SPACES and CAR RETs) is not a case shift a LOWER CASE will be output in the front of it.]

b a10

```

e17: nc 58 , ca 60 ; procedure outchar(char); integer char;
      ga a , hv s1 ; if char = <LC> V char = <UC> then case 1:= char
      ga a3 , nc 64 ; else if char ≠ <CR> ^ char ≠ <SP>
a: pm 58[case1]DXV NZ ; ^ (first V case 1 ≠ out case) then
[-1] arm a3 , hv a3 ; begin if first then begin first:= false;
      ca (a1) , hv r-1 ; outcase:= case 1 end
      bs (a1) t 60 ; else
      sc (a1) D ; outcase:= 118 - outcase;
a1: nt 118 , sy 118 ; sy(outcase)
      [outcase] [first] ; outsum:= outsum + outcase + 1; go to a3
      arm a1 , ud a4 ; end else
a3: sy [char] , ca 61 ; a3: begin sy(char); if char = <SUM> then
b3: hs e14 [see 3e37] ; treat sum else
      ca 28 , ntn(b14) ; outsum:= if char = <CLEAR> then 0
a4: ac (b14) DXt 1 ; else outsum + char + 1
b39: qq [check], hv s1 ; end outchar, check is used at 4e46;

```

e

```

<d41,
b40: tfault; ; alarm texts
>
b41: toverlap; ;

```

b a10

```

a:  qq          ; integer array inbuff[1:7];
    qq
[-3] qq
[-2] qq
[-1] qq 512      ; comment part 1 of 2a - 5a are used in Pass 1;
e18: qq          ; ENDACTIONS:
[6a] qq r-1,     ; case inbuff[7] of
                ; begin
[1] b9: qq IB [=1.39] ; end track end text: go to end text;
[2]   pa 1e7 t e13   ; end text: begin alter (input next word) to proceed
[3] e19:hv 4i [see e21] ; to: (trailing edge); go to end word end;
[4] b11:qq -1        ; undef 3: go to end word; undef 4:;
[5] b12:qq IZA [= 10.39] ; undef 5: ;
[6] b13:qq DXVN[=960.39] ; end track:; comment R:= 0;
[7]   hv (e7) Dt -7   ; end word: begin reset pointer; goto NEXT CHAR end;
                ; end endactions;
e20: arm -2 , sr b9   ; if intrack - 1 = last out in pass 1
    sr b22      NZ   ; V overlap then
    hv e29      LZ   ; go to OVERLAP ERROR;
    hs e24 , qq e1-1 ; select track(intrack); s:= lower buff - 1;
[s1] ac -2 , pm b10 ; intrack := intrack + 1;
    sc b24 X ITB ; inlength:= inlength - 1;
    bs (a2) t 39e1 ; if wordad > upper buff then
    gs a2 , ps 39e1 ; begin wordad:= s; s:= upper buff - 1 end;
    lk s1 V NTB ; if inlength > 0 then read track to (s + 1) else
    grn -2 , grn e20 ; begin intrack:= nonsense; overlap:= true end;
    mb b11 , is (a2) ; cell[wordad + 40]:= cell[wordad + 40] - 1 pos 3;
e21: ac s40 , hv (e50) ; comment mark end of track; go to end word;
[later changed to: (e19) ] ; comment initially: return;
                ; NEXT WORD:
e22:a2: pmn 39e1 t 1 ; wordad:= wordad + 1; M:= cell[wordad]; R:= 0;
<d41,pa a3 Vt a-1 ; skip line;
e23: ps r-2 , hv 20e1 ; BUFFERMEDIUM WORD: next word;
    pa 6a Vt i-6a+2LZ ; if R ≠ 0 then
e32: qq b40 , hs e16 ; alarmprint({fault});
[+2]
x pa a3 t a-1 ; comment no buffer case;
    pa 6a t i-6a+1 ;
[+1]>
    cln -6 , ck -4 ; clear MO;
    ca 63 , sr b11 ; for i:= 1 step 1 until 7 do
a3: ga a-1 t 1 ; begin Raddr:= M ^ 63; M:= M : 64;
    hv (6a) ; inbuff[i]:= if Raddr = 63 then 64 else Raddr
                ; end; go to ENDACTIONS;
e
e24: pm (s1) , dln b13 ; procedure selecttrack(trackno); integer trackno;
    ar b13 , ck 20 ; begin
    gt r1 , cln -10 ; vk(trackno : 960 + 960);
[+1] ga r2 , vk[group] ; vk(trackno mod 960)
    pm b9 X ; end;
[+2] vk [track], hr s1 ;

```



```

b a20
e25: ca 61 , hv a8 ; outinternal and check: if char = <SUM>
b14: ac[outsum]Dt 1 ; then go to out and treat sum;
      ca 28 , pa b14 ; outsum:= if char = <CLEAR> then 0 else
      ; outsum + char + 1;
e26: ca 64 , ar b11 ; outinternal: if char = 64 then char := 63;
a: hv r0 t 2 ; count:= count + 1; case count of
b17: 0.001- ; begin
[2] ck 10 ; begin R:= char pos 39;
e27: a1: gr e3-1 Vt 1 ; oword:= oword + 1; obuff[oword]:= R end;
[buff: gr b6 V ] ; comment buff: oword points to b6, b6:= R;
[4] ck 16 , ac (a1) ; obuff[oword]:= obuff[oword] + char pos 33;
a2: hv s1 [used to return] ;
[6] ck -18 , ac (a1) ; obuff[oword]:= obuff[oword] + char pos 27;
e35: hv s1 IQB ; comment used at 1e47;
[8] ck -12 , ac (a1) ; obuff[oword]:= obuff[oword] + char pos 21;
      hv s1 ;
[10] ck -6 , ac (a1) ; obuff[oword]:= obuff[oword] + char pos 15;
      hv s1 ; begin
b38: ar 15.3 D ; obuff[oword]:= char pos 9 + wordend pos 3 +
      ac (a1) , pa a ; set marks (obuff[oword]); count:= 0;
e28: hv s1 [buff] NA ; if -, NA then begin
      hs e24 IQB ; if buff output then go to next word out;
[s1] pm b23 , arm b23 ; LQB:= LB; select(outtrack);
      sr -2 X IZA ; NZC:= outtrack + intrack ^
      sr b20 IZB ; outtrack + inpass;
      arm b9 , ac b23 ; outtrack:= outtrack + 1;
a3: sc b21 X ITB ; outlength:= outlength - 1;
      [see 4a11] ;
      hv e29 LTB ; if outlength < 0 then goto OVERLAP ERROR;
      is (a1) , sk s-39 ; write track from (oword - 39);
      qq (a1) t -80 LQB ; if LQB then oword:= oword - 80;
      hv s1 NZC ; if -, NZC then
e29: qq b41 , hs e16 ; OVERLAP ERROR: alarmprint(<overlap>);
      ; end end end; return;
e30:
<d41, arm b50 , ar b9 ; next word out: rem:= rem + 1;
a4: gr b50 V LT ; if rem < 0 then begin
e46: pa a7 , hh a5 ; us(0,b6;rem + current base);
      ar 2b6 , us ; return end;
a5: hv s1 , pm b51 ; NEXT BLOCK: rep:= 0;
a6: bs 1 , hv a10 ; if -, first then begin
      arm 3b7 , il (b39) ; SENSE:
      il , arm b19 ; if status error(check) then begin
a7: bt [rep] Xvt-150 LT ; rep:= rep + 1; if 3 < rep then
a10: pa a6 , hv a11 ;
      qq b40 , hs e16 ; alarmprint(<fault>);
      sr b52 , is (b49) ; write(rewrite,paramword -
      bs s512 , ar 4b6 ; incr + (if up then 0 else 1024));
b34: us [rewr] , hh a5 ; go to SENSE first:= false;

```

```

a11:      ;
b55: can -1 , hv s1 ; if only sense then return;
      arn b51 , ar 3b6 ; if track = intrack
e31: bs [warn] , sr 13e1 ; ^ warning then
      hv e29 , LZ ; go to OVERLAP ERROR;
      arn b9 , ud a3 ; outlength:= outlength + 1; if outlength<0
b54: hv e29 , LTB ; ^ -, tape output then go to OVERLAP ERROR;
b49: bs 513 , ar 4b6 ; write(unit; paramword +
b35: us [unit] , it 512 ; (if up then 1024 else 0));
      xr (b49) , ac b23 ; block count:= block count + 1;
      bsn(b49) , ar 4b6 ; up:= -, up; current base:= buffer base +
      ar 1b6 , gr 2b6 ; (if up then 1024 else 0);
      arn b52 , ac b51 ; paramword:= paramword + incr;
      srn b53 , hv a4 ; rem:= -block length - 1; go to next word out;
>      ; comment end buffer case;
a8: hs a ; out and treat sum: outinternal(<SUM>);
e14: arn b14 , tk -5 ; treat sum:
      ar b14 , ud e37 ; char:= outsum : 32 + outsum
      ar 31 , D ; ^ 31 + 31;
      pa b14 , ga a9 ; outsum:= 0;
      hr (a) , t 2 LQA ; if -, external then go to outinternal;
a9: sy [sum] , hr a2 ; sy(char); return;

```

e

[Byte input:

Bytes output in pass 1 are packed 4 into a word. Last words on even tracks are A marked, on odd tracks A and B marked]

```

b a10
a: qqf , ; integer array byte[1:4];
  qq , ; comment f is used at END INIT, page 18;
  qq ,
  qq ;
e33: a1: pm 39e2 Xt 1 ; byte input: iword:= iword + 1;
      ga a , ck 10 ; R:= set marks (bbuffer[iword]);
      ga 1a , ck 10 ; for i:= 1, 2, 3, 4 do
      ga 2a , ck 10 ; byte[i]:= part(1) of: (R);
      ga 3a ;
b15: pa 3a Dt a ; reset counter;
      hv a2 , NA ; if -, NA then begin byte track: LQB:=LB;
e34: vk 960 , IQB ; wait for track; work length:= work length-1;
b18: btn -1 , t -1 ; if work length > 0 then begin
      hs e24 ; select track(pass input);
[s1] ac b20 , is (a1) ; pass input:= pass input + 1;
      lk s-39 , NZ ; read track to(iword - 39) end;
      qq (a1) , t -80 LQB ; if LQB then iword:= iword - 80 end;
a2: arn a , hv s1 ; Raddr:= byte[1]; return;

```

e

[The following procedure is called to examine erroneous input characters, checksums or to display the input stream on the typewriter after errors.

The procedure is used in both passes]

```

b a20
e43: xr      , mb r1      ; treat char: swap; if (Raddr ^ 63) = 63 then
[1]  ca 63    , hr s-3    ; ALL HOLES: blind return;
     xr      , it a9      ; LT:= true; text:= <char>; skip line;
e36: pa a     t a10      ; treat slow: text:= <parity>;
     ca 639   , hr s-3    ; if Raddr = 127 + 512 then blind return;
     ga a4    , gs a1     ; char:= Raddr;
     vy d43   ; select(alarm unit);
     pm s-1   X          ; if cell[s-1] ≠ 0 then
     hv a2    X          LZ ; parity or char: begin
     gr a3    , grn s-1   ; saved instr:= cell[s-1]; cell[s-1]:= 0;
a:   qq [text] , hs e38   ; mess: writetext(text);
     bs (b)   , sy 60     ; if case ≠ 0 then writechar(<UC>);
     pa a5    t -511     ; line count:= 2;
b16: a8:vy [by] , qq      ; select(saved by);
a1:  ps -1    , hr s-3    ; blind return end;

a2:  hv a7          LA   ; if marks(cell[s-1]) ≠ A then
a3:  zq              ; print: begin execute saved instr:
     [ga          V      NT] ; if LT then
     hv a          ; go to mess;
a4:  sy [char] , vy (b16) ; sy(char); select(saved by);
     ca 64      , it -1   ; line count:= line count - 1;
a5:  bt [line] , pm a3    ; if line count < 1 then
a6:  gm s-1     M        ; cell[s-1]:= saved instr;
     pm b9      , hr s1   ; marks:= 0; return end;

a7:  pm b26      , ud a6   ; check: cell[s - 1]:= saveslowaction;
     pm (b1)    DXt -62    ; marks[s-1]:= 0; insum:= insum - <SUM> - 1;
     hv a13     NPA      ; if only input sum then begin
     ga b14     , nt (b48) ; outsum:= insum - insum2;
     qq (b14)   , pa b48   ; insum2:= 0 end;
a13: tk -5      , ar b1    ; if (insum : 32 + insum)
e37: mb 31      D[see 1e14] ; ^ 31 + 31 ≠ char then
     ar e37     , nc (a4)  ;
     qq a11     , hs e38   ; writetext(<tapesum>);
     hs b3      LPA      ; if only input sum then treat sum;
     hs e45     M        ; if ready then begin A:= true; set summing end;
     pa b1      , hv a8    ; insum:= 0; select(saved by); blind return;

a9: k 63, 58, 29
tchar;
63, 62.
a10: k 63, 58, 29
tparity;
63, 62.
a11: k 58, 29
t tapesum ;
62.

```

e

b a10

```
e38: pa a      , it (s)      ; writetext: itext:= part 1(cell[s]) - 1;
a:   pm[itext] t 1          ; next word: itext:= itext + 1; M:= cell[itext];
a1:  arn a2     , cl -6      ; comment R(37:39) = 3 1;
      tk -4      , ga a3     ; next char: char:= M ^ 63; M:= M : 64;
      ca 15      , hv a      ; if char = 15 then go to next word;
a2:  ca 10      , hv s1     ; if char = 10 then return;
      ca 63      , it 1      ; sy(if char = 63 then 64 else char);
a3:  sy [char] , hh a1      ; go to next char;
```

```
e39: sy 58      , sy 29      ; display: writered; write LC; writecr;
      sy 64      , tl -30    ; M:= Raddr × .001; R:= Raddr : 1000;
      pa a7      Xt 3        ; zero:= <SPACE>;
      pt a5      , mln b17   ; for i:= 1 step 1 until 4 do
a4:  ck -10      , ga a6      ; begin
a5:  pa a6      Vt[zero]LZ   ; if R ≠ 0 then zero := <0>;
      pt a5      t 16        ; writechar(if R = 0 then zero else R);
a6:  sy 0        ; R:= entier(M × 10);
a7:  bt [i]      t -1        ; M:= M × 10 - R
      mln b12     , hv a4     ; end;
      sy 62      , hr s1     ; writeblack; return;
```

e

b a0

[The following locations overwrite Initialize pass 1.]

d a0=i

<d41,

```
b50: qq [remaining.39]      ;
b51: qq                      ; paramword
b52: qq [increment.39]      ;
b53: qq [blocklength.39]    ;

b19: qq >                   ; output status
b20: qq [pass input.39]      ;
b21: qq [outlength.39]       ;
b22: qq [last output track in pass 1.39]
b23: qq [outtrack.39]        ; [block count.39]
b24: qq [inlength.39]        ; [work end in pass 1]
b25: qq [first output block.39]
b26: qq                      ; save slowaction
```

d e9=i-1, e10=1e9, e11=-e9+1 ; first word in the stack:

[1b26] qq ; used only in pass 1

[2b26] qq ; -

d i=a0

e

b a30

```

a:  hs a3          ; table change: index:= next param;
    gi a2      , hs a3 ; type:= next param
a1:  ps -1      , ps s1 ; for s:= 1 step 1 until no of types do
    arn sa7     , nc (a4) ; if type = part 1(types[s]) then go to found;
    bsn sa8     , hh a1 ; alarmprint(<param>);
    hv a5       LZ ; found:
    tk 10       IRC ; character table[index]:=
a2:  gr[index] t e5 MRC ; types[s] x 2/10;
    arn a4      , ca 41 ; if type = replaced then begin
    ps r1       , hv a3 ; part 1[table base + index]:=
a16: pp p1      , hh a9 ; next param; caution:= true end;
    pm a17      , gm e47 ; p:= p + 1; go to take param;
    gi (a2)     , hv a16 ;
a17: pa e8      t e25 ;

a3:  pm p1      , pp p1 ; next param: M:= set marks(param[p+1]);
    hv a5       LC ; if marks = end then alarmprint(<param>);
    cln -30     LA ; if marks = underlined then M := M shift -30;
    cln -7      , ck -3 ; p:= p + 1; Raddr:= next param:= M;
    ga a4       , pi (a4) ;
a4:  bs 0       t 64 ; if Raddr > 64 then
a5:  qq b42     , hs e16 ; alarmprint(<param>)
    hr s1       ; return;

```

[character types: see also page 1]

```

a7:  qqf 37      ; normal
    qq 49       , qq -2 ; alarm
    qq 41       ; replaced
    qqf 50      , hvf s1 ; blind
    qq 18       t 512 ; skip
    qq 53       t 770 ; end
    qq 51       t 896 ; case independent

```

d a8=-i+a7+512

<d41,

```

a14: arn(c15) Dt -1 IZA ; procedure backup; begin comment JJ;
    nc d14-1 , hv a15 ; iword:= iword - 1; if iword < d14 then
    pm (c4) DXt -1 M ; begin track:= track + 1;
    ca -1 , ac c5 ; if track = -1 then begin track:= 959;
    pa c4 t 959 NB ; group:= group - 1; end;
    qq (c8) t -2 ; reltr:= reltr - 2;
    hs c5 ; read track;
    pa c15 t 38d14 ; iword:= d14 + 38 end;
a15: pm (c15) , hr s1 ; set marks(store[iword]) end backup;
>

```

[Start Init pass 1:

The parameters for the program are examined and the following actions are executed:

i<area>,o<area>: the description of the <area> is searched for and stored for later use

1 the control is transferred to the table change routine

< terminates the scanning. The current input medium is re-selected and pass 1 is started]

```

d a12=-126 ;
e40: qq (c1) t 2 ; PROGRAM START:
<d35-2, us(-31)t -96 ; if aux kind = tape then rewind tape;
> vk 960 , vk d11 ; free is treated as normal area;
lk a12 , vk 960 ; read parameter track;
<d41,pa c75 t 39a12 ; assign buffer to read internal;
pa c76 t 20a12 ;
qq (-5) t a12-d14 ;
> [pp 1a12 , ] ; p:= base for parameters;

```

```

a9:  pp 1a12 , arn p ; take param: R:= set marks(param[p]);
    hv a10      LC ; if marks = end then goto end scan;
    hv a5       NA ; if marks ≠ underlined then alarmprint(⟨param⟩);
    ca 35[1] , hv a ; if R = <1> then go to table change;
    pm p1 , nc 57 [1]; if R ≠ <i> ∧ R ≠ <o>
    nc 38[o] , hv a5 ; V marks(param[p+1]) ≠ text
    hs c1 X NC ; V search(p) ≠ 0 then
    hv a5 NZ ; alarmprint(⟨param⟩);
    arn 2c , ps (p) ; s:= Raddr; if s = <1> then
    bs s473 , is 58 ; input area:= areaword else
    gr sb29 , tl -7 ; output area:= areaword; Raddr:= type(areaword);
<d41, pp p1 ; if buffermedia then begin p:= p + 1;
    nc s-35 , hv a13 ; if Raddr = tape ∧ s = out then
    hs a14 ; begin backup;
[-1] hs a14 ; while NA do backup;
    hv r-1 NA ; backup;
    hs a14 ; if NA then get word;
    hs c15 NA ; save search on parametertrack
    vk 960 , vk d11 ;
    sk c , vk d11 ;
    arn 2c , tl -7 ; end end else p:= p + 1;
a13: xpp p1 , nc 0 ; if -, buffermedia ∧ 0 < Raddr
    hv a5 NT ; V Raddr = constant
>a6: pp p1 , nc -4 ; V Raddr = ly ∧ s = out
    nc s-59 , ca s-39 ; V Raddr = sy ∧ s = in V program bit
[-1] hv a5 LT ; then alarmprint(⟨param⟩);
    tk 12 , ud r-1 ; ignore text: p:= p + 1;
    arn p-1 , tl -6 ; if bits 0 to 3 of(param[p-1]) = -1
    ca -1 , hvm a6 ; then go to ignore text;
    bs s473 , pm p ; if s ≠ out then go to take param;
    hh a9 NB ; if marks(param[p]) ≠ number then begin
    pp p1 , cln -10 ; p:= p + 1; zerotracks:= param[p] end;
a10: vk d21 V LC ; go to take param;
    ga b36 , hh a9 ;

    lk d14 , vk 960 ; end scan: fetch catalog track;
    arnd14+d45, gr 1b26 ; R:= save work as output:= work as output;
    tl -32 ;
    tln 36 , gt b33 ; worklength:= blocks(R);
    tl 1 , nc 0 ; if worklength > 511 then
    pt b33 t 511 ; worklength:= 511;
    tl -21 , gr b24 ; outtrack:= work end:=
    tln 16 , ar b24 ; blocks(R) + first block(R);
    gr b24 , gr b23 ;

    pi 0 , arn -2 ; if current input ≠ <ly> then
    hh a11 NC ; begin
    pm -2 , hsn c3 ; select track(cell[-2]);
<d41, grn b37 , lk a12 ; read track to(text buffer);
x grn b37 , lk d14 ;
> pa e42 t e41 ; input:= input internal
a11: vk 3 , mb a11 ; end;
    ca 1 , pi 1 ; twr input:= current input = twr;
    vy d43-1 t -d43 ; select also(alarm unit);
b31: sy 64 LRB ; if twr input then writecr(2);
    gk b16 , ud b31 ; saveby:= by;
    grn e3-1 t 1 M ;
    it (r-1) , bs 78e3 ; initialize marks in output buffer;
    hv r-2 ;
    gr 39e3 V MA ; comment see next page;
    e ; go to START PASS 1;

```

```

b a40
e41: hs c27 ;
[s1] gr 79e3 MC ; input internal: Raddr:= READ INTERNAL;
hv a2 ;
[s3] ca 63 , sr b11 ; if Raddr = 63 then Raddr:= 64;
hr s1 ; return;
e48:
b32: lyn e18 V ; input external: Raddr:= lyn; skip line;
arn 10e5 , hv a28 ; reset and stop pressed: Raddr:= ten; go to TEST END;
mb r1 , pm e18 ; if (Raddr ^ 63) = 63 then
[1] ca 63 , hv e48 ; go to input external;
hr s1 X ; return;

```

[Line buffering:

If the input is taken from the typewriter, generated bytes are buffered and output in a burst after every CAR RET. This makes possible the erasing of a line until it is terminated. The action for the line termination is started if p < 0 or if the buffer is full. (it can store 80 bytes)

```

a: gr e1-1 , gs a2 ; STORE BYTE: car:= car + 1; buffer[car]:= Raddr;
bs (a) t 78 e1 ; blind CR: if car > upper bound then
sy 64 , hv r3 ; writecr else
nc 64 ; if Raddr ^ <CR> ^
bs p1 , hv s1 ; p > 0 then return;
[3] it (a7) , pt a3 ; last state:= state;
it (a23) , pt a4 ; last case 2:= case 2;
pp e1-1 , ps r ; for i:= 1 step 1 until car do
[s1] pp p1 , arm p ; direct out(buffer[i]);
it p-1 , bs (a) ;
hv (a17) t 2 ; comment store end OK and in2nd;
gi a5 V ; last in:= in; skip line;
a1: qq b44 , hs e38 ; ERASELINE: writetext({annul});
a2: ps a11 , pp 0 ; START PASS 1: p:= 0;
pa a t e1 -1 ; car:= 0;
a3: pa a7 t0 [last state]; state:= last state; comment initially 0;
a4: pa a23 t [last case 2]; case 2:= last case 2;
a5: pi 256 t 255 ; in:= last in; comment initially end OK ,-, in2nd;
; return;

```

[Test terminator:

After every underline () regardless to its type, the next character will be compared to a period (.). Should the test fail, LQA will be set to indicate that the next character must be taken from e18-2.

```

a6: hv s1 , hs (e42) ; UNDERLINE:
ps a11 , ga e18-2 ; store char:= input
pm 14 [ ]D M ; if storechar ^ <.> V case ^ 0 then
ca 59 [.] , bs (b) ; begin LQA:= true; go to TESTTYPE end;
hv a13 X IQA ;

```

```

pm a34 , am e18-1 ; STRING TERMINATED:
qq (b1) Xt 75 IZA ; insum:= insum + 59 + 14 + 2; in2nd:= false;
sy 59 LZ ; if slowaction = print then writetext(⟨<.⟩);
a7: bt [state]t -1 ; state:= state + 1;
am(a7) DXV IZB ; if state > -1 then endOK:= state = 0 else
pm a7 Vt 2 IZA ; begin state:= 2; in2nd:= true end;
a8: qq [count]Vt 1 LZB ; if endOK then count:= count + 1;
pa a23 X LZA ; if in2nd then case 2:= 0;
pa a15 t -e10+1018 ; byte count:= max allowed;
hv a17 NZB ; if -, endOK V -, twr input then begin
hv a17 NRB ; outbyte(512); go to NEXT CHAR end;
pm a8 X ; TRIPLE TERMINATION: writecr; p:= -1;
pp -1 , hs e39 ; display (count);
sy 64 , sy 64 ; writecr(2);
pm e18-1 , hv a16 ; out M (512); go to NEXT CHAR;

a9: pi 0 Xt -9 LQA ; CASE SHIFT:
a10: bt[shifts]Xt -150 LRB ; shifts:= shifts + 1; if shifts > 3
hv a24 X LRB ; ^ twr input then go to NUMERIC;
d a11=i-1 ; NEXT CHAR:
hs (a13) LPC ; if blind then execute(action[char + table base]);
e42: [-3] hs e48 NQA ; char:= Raddr:= if -, LQA then input
pm e18-2 , ud a9 ; else storechar; LQA:= false;
a34: gaf a13 V NT ; if parity V slowaction ≠ 0 then
[ f is used at e47]; begin only input sum:= false;
a12: hs e36 t a11 IPA ; treat slow end;
nc 64 [CR] LRB ; if Raddr = <CR> ^ twr input then
ca 14 [ ]V IPC ; p:= negative; blind:= false;
pp a-1a11, hv a13 ; if Raddr = <.> then begin
hh a6 IZB ; endOK:= false; go to UNDERLINE end;
ac (b1) Dt 1 LQB ; if sum then insum:= insum + Raddr + 1;
nc 13 LRB ; if Raddr = <aa> ^ twr input then
a13: pm [char] Vt e5 IPC ; begin writecr; go to PERHAPS ERASE end;
sy 64 , hv a27 ; TEST TYPE: M:= set PC(character table[char]);
nc 58 , ca 60 ; if Raddr = <LC> V Raddr = <UC> then
hv a10 ; go to CASE SHIFT;
pa a10 , ca 28 ; shifts:= 0; if Raddr = <CLEAR>
mt -1 D LZA ; ^ in2nd then Raddr:= -Raddr;
hv a22 X NPB ; if spec then begin swap; go to TREAT SPECIAL end;
a14: hv a29 LZA ; after spec: if in2nd then go to CHECK CASE 2;
hh pla11 LPC ; if blind then go to if p < 0 then blind CR
pi 0 Xt -257 ; else NEXT CHAR; endOK:= false;
a15: ncn[bytes]XVt -1 ; next out: byte count:= byte count - 1;
a16: ps a11 XV ; if byte count = 0 then alarmprint(⟨full⟩);
ar b , ud a19 ; outbyte(Raddr + case + character table base);
hv a20 NZC ; go to NEXTCHAR;
; out M: set return to (NEXTCHAR); swap;

```



```

a17: hv r0      t 2   NRB ; OUTBYTE: if twr input then go to STORE BYTE;
      hv (a)    Dt 1   ; directout: pos:= pos + 1; case pos of begin
[2] a18: gr e3-1 [oword]t 1;      begin oword:= oword + 1;
      hv s1      ;      obuff[oword]:= Raddr end;
[4] ck -10     , hh r2 ;      obuff[oword]:= obuff[oword] + Raddr pos 19;
a19: ar e5      DV      ;
      [used at 2a15] ;
[6] ck -20     , ac (a18) ;      obuff[oword]:= obuff[oword] + Raddr pos 29;
      hv s1      ;      begin
[8] ck 10      , ac (a18) ;      obuff[oword]:= set marks(obuff[oword])
b30: arm -1     Dvt 1 LA ;      + Raddr pos 39; pos:= 0; if LA then
      [used tracks] ;      begin used tracks:= used tracks + 1;
a33: pa a17     , hv s1 ;
b33: gm 2b26    , ca[length];      if used tracks := worklength then
a20: qq b43     , hs e16 ;      alarmprint(<full>);
      arm b9     , sc b23 ;      outtrack:= outtrack - 1;
      hs e24     ;      select track(outtrack);
      qq b23     , pm (a18) ;
      is (a18)   , sks -39 ;      writetrack from (oword - 39);
      qq (a18)   t -80 LB ;      if Bmark(obuff[oword]) then
      pm 2b26    , hv a33 ;      oword:= oword - 80;
      ;      end end end; return;
      ; TREAT SPECIAL:
a22: bs (a13)   t 64e5 NPA ;      if char > 64 v -, NPA then
      ps a12     , hv e43 ;      begin set return to(NEXT CHAR);
      ca 512     , hhp 1a11 ;      treat char end; if skip then go to if p < 0
      nc 896     XV LT ;      then blind CR else NEXT CHAR;
      ga a13     , hv a13 ;      if replaced then begin char:= Raddr;
      hv a28     X ;      go to TEST TYPE end; if -, case free then
      sr b       XV NZA ;      go to TESTEND; if -, in2nd then begin endOK:=false;
      ;      Raddr:= Raddr - case; go to next out end;
a29: pm b       XV ; CHECK CASE 2:
      hv a15     IZB ;      if case ≠ case 2 then
a23: ca[case2] , hv a16 ;      begin
      ga a23     , ud e12 ;      case 2:= case; outbyte(if case = 0
      arm 60[UC]D ;      then 58 else 60)
      ps a16-1 , hv a17 ;      end; go to outbyte;

a32: qq b47     , hs e16 ; SYNTAX: alarmprint(<termination>);

a24: ca 60      , hv 1a11 ; NUMERIC: if char = <UC> then go to NEXTCHAR;
      syn 29     , sy 17 [<] ; case:= numb:= digit:= 0; writeread; writechar(17);
a25: pa b       , ml b12 ; DIGIT: numb:= 10 × numb + digit;
[1a25] lyn a26 , ca 60 ; INCH: R:= digit:= lyn; if Raddr = <UC> then
      pan b      t 128 ;      begin R:= 0; case:= 128 end;
      ca 58      , pan b ;      if R = <LC> then R:= case:= 0;
      hv 1a25    LZ ;      if R = 0 then go to INCH;

```

```

      ca 16[0] , pan a26 ; if R = 16 then digit:= 0;
      is (b) , bs s511 ; if case = 0
a26: bs [digit]t 9 ; ^ digit < 9 then go to DIGIT;
      cln -10 V ; shifts:= 0;
      ck 10 , hh a25 ; char:= numb;
      ga a13 , pa a10 ; writeblack;
      sy 62 , hv a13 ; go to TEST TYPE;
      ; PERHAPS ERASE:
a27: hv a1 NZB ; if -, endOK then go to ERASE LINE;
a28: pm 513 DXV LZB ; TEST END: if -, endOK then
      ca 832 , hv a32 ; go to if Raddr = ten then SYNTAX
      hv a14 X NZB ; else after spec;
      pp -1 , hs a17 ; p:= negative; outbyte(513);
[s1] bt 159 t -1 ; for i:= 159 step -1 until 1 do
      hvn(a17) t 2 ; direct out(0);

```

[The tracks containing the bytes were output from the end of the working area backward. The order of the tracks will be reversed, thus during pass 2 the tracks will become free for output as the bytes are processed.]

```

      arm b23 , gr b20 ; pass input:= outtrack;
      gr b22 ; last track:= outtrack;
      pp (b30) , pp p1 ; p:= work length:= used tracks + 1;
a30: gp b18 , bs p510 ; test middle: if p < 2 then
      vk 960 , hh a31 ; go to reversed;
      pp p-2 , hs e24 ; p:= p - 2; select track(outtrack);
      qq b23 , lk e1 ; read track(buf 1);
      sc b24 , hs e24 ; workend:= workend - 1; select track(workend);
      qq b24 , lk 40e1 ; read track(buf 2);
      sk e1 , hs e24 ; writetrack(buf 1); select track(outtrack);
      ac b23 , sk 40e1 ; outtrack:= outtrack + 1; write track(buf 2);
a31: hh a30 , vk c63 ; go to test middle;
      lk e1 , lk 40e1 ; reversed: fetch INIT MEDIUM;
      arm a8 , nc 0 ; if count ≠ 0 ^ -, twr input then
      hs e39 NRB ; display(count);

e ; end Pass 1;

```

[Select and initialise input and output media for Pass 2:]

```

b a30
d a = e1 + c82 - c28 ; define bufferpart

      arm b28 , ga b5 ; R:= output area; kind:= Raddr;
      tl -2 , pa b16 ; saved by:= 0; if Raddr : 4 = work then
      ca1.5+d32.7, it b46 ; R:= work as output;
      arm b28 , tl -7 ;
      hv a1 NT ; if kind = sy then begin
      tk 17 , ga b16 ; saved by:= part 2(R);
      pa e8 t e17 ; external:= true; out:= outchar;
      pa a5 , hvn a12 ; go to INIT INPUT end;
a1:<-d41+1, ; if -,buffermedia then begin
      tl -25 , tln 16 ; outlength:= blocks(R);
      gr b21 , tln 16 ; outtrack:= first output block:=
      gr b25 , gr b23 ; first block(R);
a12: qq e2 , vk 960 ; go to INIT INPUT end;
x qq e2 , ga a2 ; INIT MEDIUM(R);
      tl 7 , hs e1 ; outlength:= length in init;
      pm e1 , gm b21 ; go to case kind + 1 of
a2: hhr [kind], arm -2 ; (DRUM,
[1] hh a4 , pt a11 ; DISC,
[2] hv a3 , pm 4b6 ; CARR,
[3] hv a9 , pm a13 ; TAPE);

```

```

      hv a14          NZ ; TAPE: if R  $\neq$  0 then go to label error;
      arm b28        , cm b27 ; if unitpart(input area) = unitpart(output area)
      pa b34        Vt 32 ; ^ kind(input area) = tape then
      qq b41        , hs e16 ; alarmprint({overlap}); rewrite:= 32;
      pa e44        , grn b23 ; tape output:= true; block count:= 0;
      grn b54       , hv a10 ; go to COMMON;
a3:   arm b28        , sr b27 ; DISC: disc:= true; if input area = output area
      pa e31        t 1 LZ ; then warning:= true; go to COMMON;
a4:   hv a10        , gr b25 ; DRUM: outtrack:= first output block:=
      gr b23        , pa e15 ; track in init; filler:= 0;
      hvn a12       ; go to INIT INPUT;
a9:   arm 2         DV ; CARR: M:= 1024; R:= 2 pos 9; skip line;
a10:  pm 3a         , arm 1a ; COMMON: M:= block length in init; R:= incr in init;
      gr b52        , gm b53 ; block length:= M; increment:= R;
      srn b53       , sr b9 ; rem:= -block length - 1;
      gr b50        , ar 2a ; paramword:= current block in init + incr in init
      ar 1a         , ar a16 ; + rem + 2568;
      gr b51        , arm b6 ;
      gr e27        , arm a15 ; buffer output:= true;
      gr e28        , arm 4a ; check:= check in init;
      ga b39        , tk 10 ; unit:= unit in init +
a11:  ar 16         Dt -16 ; (if disc then 16 else 0);
      ga b35        , it (b35) ; rewrite:= rewrite + unit;
      qq (b34)      , cln -10 ; filler:= block length - 39;
a12:  vk e2         , ac e15 ;
>   arm b27        , hs 40e1 ; INIT INPUT: if INIT MEDIUM(input area)  $\neq$  0
<d41, hv a14       NZ ; then go to label error;
>   pm -2          IQC ; if current input = external then
      hh a8         X NC ; go to EXTERNAL INPUT;
      pm 40e1       , gm b24 ; inlength:= length in init;
      pp 40e1       , hv e20 ; init inchar internal: end track;
e50:  hv r0         Vt e49 NQB ; if current input  $\neq$  drum then begin
<d41, vk 950       ; wait for track;
      pa 18e1       t 11e1 ; get state; comment for comparsion
      arm 18e1      , il ; with output block;
      pa 1e7        t e23 ; input:= BUFFERMEDIUM WORD end else
>[-1]pt e21       Vt e19 ;
d e49= i-e50-1    ;
[1] gp e22        , hv e20 ; begin wordad:= upperbuff; end track end;
a17:  vy (b16)      , pm r ; END INIT: select(saved by); LB:= false;
      pp 79e2       , hs e34 ; byte track; LB:= true;
      gp e33        , hs e34 ; iword:= e2+79; byte track;
b36:  a7: bt [zero] t -1 ; for i:= 1 step 1 until zerotracks do
      pa a6         Vt 240 ; for j:= 1 step 1 until 240 do output(0);
a5:   pi 8 [LQA], hv e6 ; go to START PASS 2;
a6:   bt 240        t -1 ;
      ps r-2        , hvn(e8) ;
a8:   hv a7         , vy 1 ; EXTERNAL INPUT: select(twr input);
      ac (b16)      D ; saved by:= saved by + current input;
      pp e13        , ca d17 ; if current input = reader ^ NRB ^
b37:  lyn          D NRB ; -, internal input in Pass 1 then lyn;
      gp 1e7        MA ; place external input instructions;
      pm b32        , gm e7 ; go to END INIT;
      hv a17       ;

```

```

<d41,
a13: qq -1.2+1.23-1.27 ; mask kind ^ unit
a14: qq b45 , hs e16 ; label error: alarmprint(<label|);
a15: hv e30 ; buffer output
a16: qq 2568.39 ; 1543 + 1024 + 1
>
b27: qq -1.2+d17.19 ; input areaword: reader
b28: qq -2.2+32.19 ; output areaword: punch
.
d b29=b27-57, b46=1b26-b28 ;

b42: tparam; ; alarm texts
b43: tfull; ;
b47: ttermination; ;
b44: k 29
      tannul;
63, 62. ;
<d41, ;
b45: tlabel; ;
>
e ; end init pass 2

< i-c27+10, ilength
>
< i-c48 , ilength1
>

```

```

d i=i+39, d=k-d1          ; d = no of tracks
b k=d42, i=0, a10         ;
d a1=d19-960              ; a1= group no for image
d a=d, <d35, a=2 >        ; a = no of blocks

<d39,                      ; if aux only then
d i=d2                     ;
  hs 1                     ; begin
  hv a5                    ;
<d36,                      ;
  tres;                   ; (if aux reserved then
  qqf a.39                 ; res; no of blocks,
x<d39;                     ; else
  tset;                   ; set, aux kind, no of blocks,
  qqf d35.39               ; typein)
  qqf a.39                 ;
[STOP, SUM] ibase, edit
s
[STOP, CLEAR]
x<d39,                      ;
  qq 39,                  ; concat p i d 0,
  qq 57,                  ;
  qq 52,                  ;
  qqf                     ;
  tedit;                  ; edit, spec <
  qqf d.9+e40.19+e4.29    ;
  qqf,                    ;

a5: hs 1                   ;
  hv a6                   ;
  tmove;                  ; move, b load place, edit <
  qq 50,                  ;
  qqf d.23+d1.39+a1.29-a1.33 ;
  tedit;                  ;
  qqf,                    ;

a6: hs 1                   ;
  hv a7                   ;
  tsetsum;                ; setsum, edit<
  tedit;                  ;
  qqf,                    ;
d d2=i                     ; end else
a7: hsf 2                  ;
x                           ;
d i=d48                    ;
  qq d35.2+11.7+d36.5+d.23+d1.39, ; load to primitive catalog;
  qq,                     ;
  tedit;                  ;
  qq d.9+e40.19+e4.29     ;
d d48=i                    ;
  qqf                     ;
>                           ;
d d1=d+d1                  ;
e                           ;
e                           ;
e                           ; end of outermost block
[STOP, SUM]c iedit
s

```

```

b k=d1, i=40d13, a10      ; begin exit
a=80d13                    ; define base of image track 0
hv a1      , vk d16        ; 0 ENTRY FROM PROGRAMCALL: goto bypass;
a2h: sk a      , it 1      ; 1 READ IMAGE: these instructions are
vk d16      , it 40        ; 2 executed in cell 0-6; write image track 0;
lk 0        , it -1        ; 3 read image track 1-24;
bt 23       , hh ra2       ; 4 read image track 0;
vk d16      , lk 0         ; 5 restore group and wait for drum;
a1: vk d19    , vk d16      ; bypass: read image track 0;
[ this instruction will usually restore group. In step execution
  group will be restored in the instruction on image track 0.]
lk a      , vy 512         ; inhibit by;
vk 25d16   , ps 9          ; init cells:= 9;
pm e      , arn 8a         ;
cm e14     , NC            ; top init:= 9;
pp 9       , hh a3         ; if core[8] ≠ vy 1 t 511 v
pm a4      , arn 9a        ; core[9] ≠ qq , hv/hh (rx) then
cm 9d12    , LA            ; goto init;
pp 9       , hh a3         ;
ps 6       , pp 7          ; init cells:= 6; top init:= 7;
a3h: gt 9d12 , ps s-1       ; std(10-19):= core[9] (10-19);
pm pd12    , t -1 IRC      ; init: for i:= top init - 1 step -1 until
gm pa      , t -1 MRC      ; top init - init cells do
bs s       , hh a3         ; core[i]:= std[i];

pm 9a      , X IRB         ; s:= 0; RB:= marks(core[9]);
ga 9d12    , arn 1d13      ; std(0-9):=core[9](0-9); if no params then
hvn a5     , LC            ; begin Raddr:= 0; goto set core 9 end;
pa 9d12    , ca 56         ; std(0-9):= 0; if param 1 = h then
arn 2d13    , is 1         ; begin Radr:= param 2; RC:= 11 end else
pi s2      , tk 30        ; begin Radr:= param 1; RC:= 10 end;
hv c58     , LC            ; if not number then
hv c58     , NB            ; goto param alarm;
a5: lk 1000 , ar 9d12      ; set core 9: read image track 25 and registers;
                                Raddr:= Raddr + std(0-9); core[9]:= if RB then
ar 7d12     , LRB          ; qqf Radr, hh (rx) else
gr 9a       , MRC          ; qq Radr, hv (rx);
vk (c5)     , vk (c4)      ; select(exit track);
ps (7)      , arn 1        ; restore s;
ga 6a       , arn 2        ; prepare restore group;
vy (2)      , t 512        ; restore by 1-9;
gt 6a       , arn 11       ; prepare restore track;
pm 8         , IRC         ; restore R00-39; RA:= 0; RB:= overflow;
tl 39       , pm 6         ; restore M and marks; restore p;
pp (12)     , hv c59       ; goto set 0; [page 6 of help]

a4: qq -1.39+1.19-1.33    ; mask for checking cell 9 of image
a:                                     exit length

```


[STOP, CLEAR]

[15.6.67] [move/begin program, init output] [page 1]

b k=d1, i=40d13, a50, b50, e40

d e11=0, e18=0 ;
 <d41,<d53-399,<-d53+401, ; e18= disc with 400 words blocks available
d e18=1 > ;
 <d41,e11=-e18+1 > ; e11= other disc available
d e10=c74-400 ; core buffer base

[40d13] [input areaword]

b47: pp d13-1 , hs e19 ; p:= param base;
b4: tk 10 , gt b12 ; take input param;
 <d41,
b38: hv [i1] , qq > ; [init input action]
[s3] [output areaword]
b37: hs e19 ; take output param;
b5: tk 30 , ga b13 ;
 <d41,
b39: qq , hv [i0] > ; [init output action]
[s3]
 <d35-2,us(-31) t -96 > ; if aux kind = tape then rewind tape;
a43: pp p1 , pm p1 ; if kind (param[p+1]) ≠ end then
 hv a45 LC ; begin
a46: mln b8 XV LB ; if kind(param[p+1]) ≠ numb v
a37: qq b44 , hs c24 ; kind (param[p+2]) ≠ end v
 sc b4 ITA ; 40 × param[p+1] > inlength then
 hv a40 LTA ; alarmprint(<length>) else
 gr b4 , grn a46 ; inlength:= 40 × param[p+1]
 <d41,hv a43 > ; end;
a45: arm b37 , vk 960 ; fetch(INIT MEDIUM);
 vk c63 , lk d14 ;
 qq e10 , vk 960 ;
 pa 3d14 , hs d14 ; if INIT MEDIUM (out areaword) ≠ 0 then
 <d41,hv a42 NZ ; alarmprint(<label>);
 hv b39 ; go to init output[kind(out areaword)];
>
e20: arm 1c , gr b1 ; init drum output:
 <-d41+1,
 gr b14 , vk c63 ; outbase:= track; go to INIT INPUT;
x gr b14 , hv a32 ;
d a31 = c82-c28+d14
e21: ; init disc output:
 <e18,arm 2a31 , ar 1a31 ; if e18 then begin
 ar b45 , gr b24 ; outbase:= cur block + 1;
 pp (4a31) , it p ; check:= check block;
 pt b21 , it p ; unit:= check block;
 pt b22 , hv a32 ; go to INIT INPUT end
x e11, ; else begin
 arm a17 , ar b10 ; block length:= d53;
 gr b26 , arm b25 ; incr:= 1 pos 21; paramword:=
 gr b27 , ar 2a31 ; incr + cur block + 1550;
 ar b45 , gr b28 ;
 it (4a31) , pt b29 ; unit:= check:=
 it (4a31) , pt b30 ; check block;
 hv a32 ; ro to INIT INPUT end;
x d41,
e22: arm 2a31 , ar1 a31 ; init carr output:
 ga b28 , hv a32 ; reel and block:= cur block + 1; go to INIT INPUT;
e23: arm 4a31 , gt b18 ; init tape output:
 ck -10 , gt b17 ; check:= check block; unit:= read block;
 grn b31 ;
 pm d13 , gm c ; save search on parameter track;
 vk 960 , vk d11 ;
 sk c , vk d11 ;


```

a32:<d41, vk c63 > ; INIT INPUT:
      lk d14 , vk 960 ; fetch(INIT MEDIUM);
      qq e10 , pa 3d14 ; medium:= false;
      arn b47 , hs 2d14 ; INIT MEDIUM(input areaword);
<d41, hv a42 NZ ; if R ≠ 0 then alarmprint(<label>);
      pm b37 X ; Raddr:= kind(out areaword);
      tl -7 , hv b38 ; go to init input[kind(input areaword)];
>
e24: pm 1c , gm b0 ; init drum input: inbase:= track;
<d41, nc 0 , hv a33 ; if Raddr = drum ∧
> arn b1 , sr b0 ; inbase < outbase then
      hv a33 LT ; begin
      pm b7 , gm b6 ; reverse:= true;
      grn b7 , pm b4 ; backward:= 1; forward:= 0;
      dln b8 , ac b0 ; R:= inlength : 40; inbase:= inbase + R;
      ac b1 , pm b1 ; outbase:= other end:= outbase + R;
      gm b14 MB ; end;
<d41, hv a33 ; go to MOVE ON;

e25: pm 2a31 X ; init disc input: inbase:= cur block + 1;
      ar b25 , gr b23 ;
      xr , ca 1 ; if Raddr = disc ∧
      xr , sr b24 ; inbase < outbase ∧
      ar b45 , pm b4 ; outbase <
      hv a34 NT ; inbase + inlength : block length
      dln b10 , tk 18 ; then alarmprint(<overlap>);
      ar b23 , sr b24 ;
      hv a41 NT ;
>a34: ; if e18 then begin
<e18, pp (4a31) , it p ; check:= check block;
      pt b19 , it p-16 ; unit:= check block - 16;
      pt b20 , arn b23 ; transfer(check-16, inbase);
      il p-16 , hv a33 ; go to MOVE ON end;
<d41,
e26: pa c66 t c-2 ; init carr input:
      pa c67 t c-3 ; prepare get word;
      pa c68 t c-1 ;
      pa c73 t c-1 ; go to MOVE ON;
      hv a33 ;
e27: nc 3 , hv a35 ; init tape input:
      arn b17 , ck 10 ; if Raddr = tape ∧
      nc(4a31) , hv a35 ; output unit = input unit then
a41: qq b43 , hs c24 ; alarmprint(<overlap>);
a35: arn 4a3 , gt b16 ; check:= check block;
      ck -10 , gt b15 ; unit:= read block;
      arn b48 , ud b16 ; transfer(param);
>
a33:

```

```

arn b5 , sr b4 ; MOVE ON: if outlength < inlength ^ -, tape output then
b31: hv a40 , LT ; alarmprint(<length>);
srn b4 , pp r ; next action: input;
hv a40 , LZ ; if inlength = 0 then alarmprint(<length>);
gr b4 , gr b5 ; inlength:=outlength:= -inlength;
pi 0 , hv e13 ; LZA:= false; go to SWITCH STATE

```

[take input param, take output param:]

```

e19: gs a36 , hs c52 ; take param:
hv a37 , nc 50[b] ; number: go to ERROR; single: if R = <b> then skipline;
hv a37 , hv a38 ; end list: go to ERROR; text: go to store word;
pp p1 , arn p1 ; p:= p + 1; R:= param[p+1];
hv a37 , LC ; if kind(R) ≠ numb then
hv a37 , NB ; ERROR: alarmprint(<param>);
a38: hv a37 , LT ; store word: if kind(R) < 0 then go to ERROR;
ga b46 , tl -2 ; kind:= part 1(R);
ca 1.5+d32.7 t 512 ; if R = work area ^ take output then
arnd14+d45, hh r1 ; R:= work as output;
tl 2 , gr (a36) ;
<d41,tl -7 ; cell[s]:= R;
ga a39 , tl -25 ; M:= blocks(R);
a36: ps [s] , tln -23 ; cell[s+2]:= set marks(medium table
a39: arn[kind] t e30 ; [kind(R)]);
gr (s2) D ; store input or output action;
ud s1 , ud s1 ; M:= M mult
mln b8 [40] , NC ; (if NA ^ NB then 40 else
mln b9 [400] , LC ; if LA ^ LB then 400 else
mln b10[d53] , LA ; if LA ^ NB then disc block length else
tl 9 [512] , LB ; if NA ^ LB then 512 else not possible);
gm s1 , hv s3 ;
xa36: ps [s] , tl -7 ; cell[s+1]:= M;
nc 0 , hv a37 ; return;
tl -25 , tln -23 ;
gm s1 , hv s2 ;

```

>
[constants:]

```

<d41,
b8: qq 40.39 ; track length
b9: qq 400.39 ; block length, tape
b10: qq d53.39 ; - - , disc
b25: qq 1.21 ; block increment, disc
b45: qq 1550.39 ; input buffer length

```

```

b41: tfault; ;
b42: tlabel; ;
b43: toverlap; ;

```

```

a42: qq b42 , hs c24 ;
xb40:

```

```

>
b44: tparam; ;
b49: tlength; ;

```

```

a40: qq b49 , hs c24 ;

```

[drum input description]

```

b0: qq [input base.39]
[p-1]lk (b2) t 40 ; input instruction
e0: qq b4 , hv e13 ; input length, endaction: SWITCH STATE;
[p1] hvf a0 ; next action: OUTPUT [fmark]

```

[drum output description]

```

b1: qq [output base.39]
[p-1]sk (b2) t 40 ; output instruction
e1: qq b5 , hv e14 ; outlength, endaction: EXIT
[p1] ; next action: INPUT [no f mark]

```

[drum I-O driver:]

```

a0: pa a1 t 9 ; DRUM:
<d41,pa b2 t e10-40 ; for i:= 1 step 1 until 10 do begin
xb2: pa[iword]Dt e10-40 ; iword:= buf base - 40;
>
a2: arn b6 , sc p-2 ; base:=base - backward;
pmm p-2 , hs c3 ; else begin select track(base);
arn b7 , ac p-2 ; base:= base + forward; iword:= iword + 40;
arn(p) , ud p-1 ; execute read or write end;
ar b8 , gr (p) ; R:= length:= length + 40;
a1: bt [i] Vt -1 LT ; if length > 0 then go to wrapup drum;
<d41,tk 30 , hv a4 ; end;
x hv p ;
> hv a2 ;
e13:b12: pm p1 , pp e0 [in]; SWITCH STATE: p:= if Bmark(cell[p])
b13: pp e1 [out] LB ; then output descr else input descr;
vk 0 , hv p1 ; go to action[p];
<d41,
a4:b2: qq [iword] t 40 ; wrapup drum: iword:= iword + 40;
[1b2]sc b2 , ps (b2) ; wrapup: for j:= iword - R step 1 until
[-2] can s-400e10 , hvp ; buf top do buf[j]:= 0;
grm s M ; go to end action[p];
ps s1 , hv r-2 > ;
b6: qq ; integer backward,
b7: qq 1.39 ; forward;
<-d41+1,b8:>
b11: qq 1.39 ; track length no buffermedium case
b14: qq[first out block.39]; also block count

e14: srn r , hs c2 ; EXIT: get catalog track;
arn b1 , pm b14 ; R:= output base; M:= set marks (other end);
qq X LB ; if LB then swap;
b46: pi [kind] , hv c74 ; in := out kind; go to ADJUST SPECIAL;

```

[tape input description]

<d41,qq e10.9+400.19+1550.39

b48: qq 400.19+1550.39

b15: qq b4 , il ; input length, check

b16: qq b6 , il ; increment = 0, transfer B < -T

e2: hv e13 , il 0 ; endaction: SWITCH STATE; transfer F < -B

[p1] qqf 0 , hvf a5 ; next action: OUTPUT

[tape output description]

[p-4] qq e10.9+400.19+3100.39

qqf 400.19+3100.39

b17: qq b5 , il ; out length, check

b18: qq b6 , ps ; increment = 0, transfer T < -B

e3: hv a23 , us 0 ; endaction: TERMINATE TAPE; transfer B < -F

[p1] qq -1 , arm b7 ; block count:= block count + 1;

ac b14 , hv a5 ;

>

[disc input description]

<e18,qq e10.9+400.19+1550.39

b23: qq 400.9 [+input base.21] + 1550.39

b19: qq b4 , il ; input length, check

b20: qq b25 , il ; increment, transfer B < -D

e4: hv e13 , il 0 ; endaction: SWITCH STATE; transfer F < -B

[p1] qqf 0 , hvf a5 ; next action: OUTPUT

[disc output description]

qq e10.9+400.19+3100.39

b24: qq 400.9 [+output base.21] + 3100.39

b21: qq b5 , il ; outlength, check

b22: qq b25 , us ; increment, transfer D < -B

e5: hv a28 , us 0 ; endaction: TERMINATEDISC; transfer B < -F

[p1]

>

[disc-tape-I-O driver]

<d41,

a5: qq -1 V ; SENSE: only sense:= true;

e12: ps s1 , gs e15 ;

pm p1 X ; DISC: TAPE:

nc 0 , hv a8 ; if first out then begin first out:= false;e16: pa a6 , hv a7 ; go to TEST; rep:= 0; go to SENSE;a6: bt 0 t -150 ; ERROR: rep:= rep + 1; if 3 < rep thena26: qq b41 , hs c24 ; alarmprint(<fault|);arm p-3 , ud p-1 ; transfer(param);

us s32 LB ;

a7: arm b35 , ud p-2 ; TEST:

il , arm b34 ; if sense status (check) < 0 thenhv a6 LT ; go to ERROR;e15: ncn 0 , hv (r) ; if only sense then return;arm(p-1) , ac p-3 ; INIT TRANSFER: param:= param + incr;a8: arm p-4 , ud p ; transfer to or from core (param 1);arm p-3 , ud p-1 ; transfer (param);

us s LB ;

arm(p-2) , ar b9 ; length:= R:= length + 400;

gr (p-2) , pa p1 ;

hv e13 LT ; if length < 0 then go to SWITCH STATUS;pa b2 t 400 e10 ; if word:= buf base + 400; go to wrapup;

tk 30 , hv 1b2> ;

[other input driver]

```

<d41,
e6: hv e13 ; endaction: SWITCH STATE;
[p1] paf b2 t e10 -1 ; next action: output;
    pa a9 Vt -399 ; iword:= buf'base + 1;
a10: gr b4 ; for i:= 1 step 1 until 400 do
a9: bt -399 t 1 ; begin
    hv e13 ; iword:= iword + 1;
    hs c71 ;
    hv a44 NZ ; buf[iword] := get word;
    qq IRC ; comment including marks;
    gm (b2) t 1 MRC ; if R ≠ 0 then alarmprint(⟨fault|);
    arm b4 , ar b11 ; inlength:= inlength + 1;
    hv a10 LT ; if 0 < inlength then begin iword:= iword + 1;
    hv (b2) DVt 1 ; go to wrapup end end;
    ; go to SWITCH STATE;

```

[other output driver]

```

d e7= i-1
[p1] pa b2 t e10 -1 ; next action: INPUT;
    pa a11 t -399 ; iword:= buf'base + 1;
a11: bt -399 t 1 ; for i:= 1 step 1 until 400 do
    hv e13 ; begin
    pm (b2) t 1 IRC ; outword(buf[iword]);
    hs a12 ; outlength:= outlength + 1;
    arm b5 , ar b11 ; if 0 < inlength then
    gr b5 ; go to END OTHER;
    hv a11 LT ; end;
    hv a27 ; go to SWITCH STATE;

a12: gm a15 MRC ; outword: loc:= R;
    arm b11 , ar a16 ; bufad:= bufad + 1; us(loc, bufad + 3100);
a14: us , gr a16 ;
    sr b26 ; if blocklength - 1 < bufad then return;
    hv s1 LT ; rep:= 0;
    pa a13 ; next block:
a13: bt 0 t -120 ; rep:= rep + 1; if 3 < rep then
a44: qq b41 , hs c24 ; alarmprint(⟨fault|);
b29: arm b28 , us 7 ; write block(unit,paramword);
b30: arm b35 , il 23 ; if sense(check) < 0 then
    il , arm b34 ; go to next block;
    hv a13 LT ;
    arm b27 , ac b28 ; paramword:= paramword + incr;
    arm a17 , hh a14 ; bufad:= -1; return;

a15: qq ;
a16: qq a15.9+1.19+3099.39 ;
a17: qq a15.9+1.19+3099.39 ;
b26: qq a15.9+1.19+3611.39 ; block length + core[a17]
b27: qq 1.9 ; increment
b28: qq 1.19+3100.39 ; paramword
>

```

[endactions for buffermedia]

```

<d41,
b34:   qq           ; status word
b35:   qq b34.9+1.19 ; get status word
b36:   qqf15.5+15.11+15.17+15.23+15.29+15.35+3.39,
[1b36] qq b36.9+1.19 ;
[2b36] qq 1.19       ;

a23: vk 960 , hs e12 ; TERMINATE TAPE: SENSE;
     arn 1b36 , us    ; transfer filemark to buffer;
     pm 2b36 X      ; param:= write filemark;
     gr p-3 , ud p-1 ; transfer(param);
     us s , hs e12  ; SENSE;
     vk d11 , ud p-1 ; rewind;
     us s64 , lk c   ; fetch search from parametertrack;
     vk d11 , vk (c5) ; fetch catalog track;
     vk (c4) , lk d14 ;
     vk (c4) , hs a20 ; backup;
[-1] hs a20 ;
     hv r-1 NA ; while NA do backup;
     hs a20 ; backup;
     hs c15 NA ; if NA then get word;
     pm b14 , tl 23 ;
     arn(c15) , ck 8 ; store[iword]:= store[iword] ^
     tl 16 , ck 16 ; 8m16016m V (block count pos 23);
     gr (c15) , hs c8 ; sum track; write track;
     sk d14 , hv b46 ; in:= out kind; go to ADJUST SPECIAL;

a20: arn(c15) Dt -1 IZA ; procedure backup; begin comment JJ; ...
     nc d14-1 , hv a21 ; iword:= iword - 1; if iword < d14 then
     pm (c4) DXt -1 M ; begin track:= track + 1;
     ca -1 , ac c5 ; if track = -1 then begin track:= 959;
     pa c4 t 959 NB ; group:= group - 1; end;
     qq (c8) t -2 ; reltr:= reltr - 2;
     hs c5 ; read track;
     pa c15 t 38d14 ; iword:= d14 + 38 end; ...
a21: pm (c15) , hr s1 ; set marks(store[iword]) end backup;

a27: arn a16 , sr a17 ; TERMINATE OTHER:
     psn a27-1 V NZ ; while bufad ≠ -1 do outword(0);
     arn b28 , hh a22 ; go to EXIT;
     hv a12 X IRC ;

>
<e18,
a28: hs e12 ; TERMINATE DISC: SENSE;
a22: arn b24 , ck 10 ; go to EXIT;
     tl -28 , ar b11 ;
     gr b1 , hv e14 ;

>

```

<d41,

e30:

[Table format:

qq init input.9+init output.19+input driver.29+output driver.39
+ block length code.41

]

[0] qq e24.9+e20.19+e0.29+e1.39 ; DRUM

[1] qq e25.9+e21.19+e4.29+e5.39, ; DISC

[2] qqfe26.9+e22.19+e6.29+e7.39 ; CARR

[3] qqfe27.9+e23.19+e2.29+e3.39, ; TAPE

>

d i=i+39, d=k-d1 ; d = no of tracksb k=d42, i=0, a10 ;a1=d19-960 ; a1= group no for imagea a=d, <d35, a=1 > ; a = no of blocks<d39, ; if aux only thend i=d2 ;hs 1 ; beginhv a6 ;tmove; ; move, b load place, move <qq 50, ;qqf d.23+d1.39+a1.29-a1.33 ;tmove; ;qqf, ;a6: hs 1 ;hv a7 ;tsetsum; ; setsum, move <tmove; ;qqf, ;d d2=i ; end elsea7: hsf 2 ;x ;d i=d48 ;qq d35.2+11.7+d36.5+d.23+d1.39, ; load to primitive catalog;qq, ;tmove; ;d d48=i ;qqf ;

> ;

d d1=d+d1 ;e ;e ;

; end of outermost block

[STOP, SUM]o imove

s

[21.9.1967

(10)

PRINT, PAIR page 1]

<-d55+1, i version

s
>

[STOP, CLEAR]

b k=d1, i=40d14, b67

d b33=d14

```

      b a39
[40d14] pp -1      vk 960      ; procedure get param track;
      vk d11      lk 0        ; begin read param track to cell 0;
      vk 0        arn p2      ; if first param=exit then goto HELP;
      sy 58      sy 64      ; LC; CR;
      hv -9      LC        ;
      pa b16      hr s2      ; end get param track;
b53:      ; ENTRY PRINT:
<d35-2, us (-31)      t-96 > ; rewind PRINT
b0:      qq 0      hs 40d14 ; next print list: get param track
b60:      sy 59      sy 0      ; entry from PAIR:
      a0: pp p1      arn p1      ; get first param:
      hv -9      LC        ; if exit then goto HELP
      hv a8      LB        ; if number then alarm({<param>});
      hv a5      LA        ; if single then goto single;
      pa a1      t e6      ; text:={<undef>};
      pmm c69 DX IZA      ;
      gp 40d14      hs c2      ; store p; search;
      pa a1      te7 NT      ; if R<0 then text:={<catalog>};
      hs c24      NZ        ; if R=0 then alarm({<text>});
      a1: qq 0      arn p1      ;
      tl -6      ca -1      ;
      pp p1      hh a1      ; count p to end of name
      it 1      ;
      qq (40d14)      hs c23      ; writetext(name);
      a2: arn 2c      gr b40      ; set: set actual area;
      tl -7      ga b8      ; set kind;
      ca 0      hv a15      ; goto kind 0;
<d41,      ca 1      hv a16      ; goto kind 1;
      ca 2      hv a19      ; goto kind 2;
      ca 3      hv a17      ; goto kind 3;
>      qq e8      hs c24      ; alarm({<kind>});
      a5: nc 50      hv a4      ; single: if -,b then goto test image;
      pp p1      arn p1      ;
      hh a2      LB        ; if base then goto set;
      arn a39      gr b40      ; act.area:=buffer descrip
      pa b8      ; kind:=0;
      a4: arn b40      sr a39      ; test image and buffer:
      hv a6      NZ        ;
      qq a32      hs c23      ; writetext({<buffer>});
      arn a38      sr e      ;
      gr b44      it 1      ; blocklength:=4096;
      a6: pa b22      pp p-1      ; boobuf:=act.area=bufferword;

```



```

arn b39      sr b40      ;
it 1         LZ         ;   booimage:=act.area=image area;
pa b0        arn b0      ;
ar b22       ga b3       ;   boobufVimag:=boobufVbooimage;
a7: pp p1     arn p1      ; next param:
ca 0         hv a8       ;   if 0 then alarm(⟨⟨param⟩⟩);
ca 36        hv a28      ;   if m then goto set group trim;
ca 35        hv a25      ;   if l then goto char per line;
ca 22        hv a24      ;   if w then goto words per line;
pa b11       ca 32       ;   extra:=false;
pa b4        hv a7       ;   if - then addr print:=false;
ca 39        ;
psn b27      IZC        ;   if p then
                        ;   begin ZA:=ZB:=true; form:=program end;
ca 41        psn b13     ;   if r then form:=real;
ca 54        psn b12     ;   if f then form:=fractional;
ca 57        psn b10     ;   if i then form:=integer;
ca 55        psn b14     ;   if g then form:=group;
ca 19        psn b16     ;   if t then form:=text;
ca 51        hv b17      ;   if c then goto PRINT CONTROL REGISTERS
gs b9        V         LZ ;   set form;
a8: qq e5     hs c24     ;   if wrong param then alarm(⟨⟨param⟩⟩);
a9: pp p1     arn p1      ; after single:
hv a10       LC         ;   if endlabel then goto empty interval;
hv a11       LB         ;   if number then goto number;
hv a10       NA         ;
ca 49        hv b18      ;   if a then goto PRINT ARITH REGISTERS
ca 37        hv a29      ;   if n then goto no relative;
ca 23        hv a30      ;   if x then goto extra precission;
a10: grn b45   grn b46    ; empty interval:
grn b47       pp p-1     ;   a:=bi:=bj:=0; p:=p-1;
can(b3)       hv r4      ;   if boobufVimage then
bs (b0)       pm a37     ;   R:=if booimage then 1023
bs (b22)      pm a38     ;   else 4095;
a3: gm b48     hv a13     ;   else R:=blocks×blocklength-1;
pmm b40       tl 7       ;
tln -23       arn e      ;   length:=R;
ml b44        hv a3      ;
a11: arn p1    gr b49     ; number:
bs (b22)      hv a14     ;   if boobuf then goto set buffer interval;
ca 0          hv a12     ;   if a=0 then goto no change;

```

```

ck 20      , nc 0      ;
hv a12     ,          ; if c $\neq$ 0 then goto no change;
ck -10     , mb a37    ;
ck 10      , ac b49    ; c:=a;
a12:arn b49 ,          ; no change:
ck -30     , mb a37    ;
xr         , mln b44   ;
gm b45     , arn b49   ; a:=(a*blocklength+b):blocklength;
ck -20     , mb a37    ;
gr b47     , ar b45    ; bj:=(a*blocklength+b)mod blocklength;
xr         , dln b44   ; bi:=b;
gr b45     , gm b46    ;
arn b49    , ck -10    ;
mb a37     , sr b45    ;
gi r3      , X        ;
mln b44    , X        IZA ;
ar 512     , D        NZA ;
pi 0       , X        ;
arn b49    , gm b49    ;
mb a37     , sr b46    ; length:=(c-a)*blocklength+d-bj;
ar b49     , gr b48    ;
a13:arn b45 , ar b40   ; end test interval:
gr b50     ,          ; mod area:=actual area+a;
arn p2     , gp 40d14  ; store p;
hv b2      ,          LC ; if -,bmarked then goto INTERVAL
hv b2      ,          NB ;
qq (40d14) , t 1      ; else bi:=param;
gr b47     , hv b2     ; goto INTERVAL;
a14:tk -20 , gr b47    ; set buffer interval:
gr b46     , ar b40    ; bi:=bj:=b;
gr b50     , arn b49   ; mod area:=area+b;
tk 20      , ck -20    ;
sr b46     , gr b48    ; length:=d-b;
grn b45    , hv 2a13   ; a:=0;
a15:pmn b40 , tl 23    ; kind 0:
tln -23    , dln c11   ;
gm b49     , ar c11    ;
qq b43     , hs b30    ; print group number;
arn b49    , hs b36    ; print track number;

```

```

      pm a34 , gm b44 ; blocklength:=40;
      hv a0 ; goto get first param;
<d41, 124/4/12b ;
      a16:pa a21 ta16-2 ; kind 1: set layout;
      pmn d53 DX ;
      cl 10 , hh a20 ; blocklength:=disc block length;
      a17:pa a21 t a18 ; kind 3: set layout;
      a18:pm a35 , hh a20 ; blocklength:=400;
      124/4/5/7b ;
      a19:pm a36 it a23 ; kind 2: set layout;
      a20:pa a21 gm b44 ; blocklength:=512;
      it (b14) , pa a22 ; store group trim;
      a21:it 0 , pa b14 ; layout group trim
b1: arn b40 , hs b14 ; print various
      a22:it 0 , pa b14 ; reset group trim;
      a23:sy 59 , hv a0 ;
      124/4/2/6b ;
      4b ;
xb1:> ;
      a24:pa a27 t b7 ; words per line:
      pt b5 , hv a26 ; boochar:=false; addr:=words per line;
      a25:pt a27 t b6 ; char per line:
      pt b5 t 1 ; boochar:=true; addr:=char per line;
      a26:arn p2 , tk 30 ;
      hv a8 NB ; if -,number then alarm({<param});
      a27:ga 0 ; addr:=number;
      pp p1 , hv a7 ; goto next param;
      a28:it p1 , pa b14 ; set group trim:
      pp p1 , arn p1 ; set addr first param-1
      hv r-1 LB ; skip b-marked
      pp p-1 , hv a7 ; goto next param;
      a29:arn b27 D IZB ; no relative: ZB:=false;
      nc s , hs a8 ; if form+program then alarm({<param});
      hv a9 ; goto after single;
      a30:arn s D ;
      ca b12 , hv r3 ; extra precision:
      ca b13 , hv r2 ; if form+r\form+f\form+i then
      nc b10 , hv a8 ; alarm({<param});
      ga b11 , hv a9 ; extra:=true; goto after single;
a32:tbuffer; ;
a34:40 ;
a35:400 ;

```

```

a36:512 ;
a37:1023 ;
a38:4095 ;
a39:qq b33.9 + 1.19 ; buffer descrip
e ;
b a14 ; INTERVAL:
b2:   arn b46   tk 20 ; set addr first cell
      gt b26   pmn b50 ;
      can(b8)   hs b21 ; if kind=0 then get track;
<d41, can(b8)   hv a2 ;
b62:  pp c-1    vk 960 ; set core base;
      vk c63    lk 0 ;
      vk 1c63   lk 40 ; read init. med. to core;
      vk 0      grn c81-c28; clear label check;
      grn b49    pa 3 ;
b61:  xr 40      pmc80-1c28; set buffer base;
      gmc80-1c28, hs 4 ; init. medium;
      pm c29-c28, pp (b62) ;
      hsn b37    sr b46 ;
      hv a2      LZ ; skip first bj words
      hs -26     ;
      srn e      ar b49 ;
      gr b49     hh r-4 ;
> a2: pa b23     t1 ; first:=true;
b63:  [hv b65     ; PAIR first run instr ]
      [hv b66     ; PAIR second run instr ]
      a3: pa a9    pa a11 ; words left:=char left:=0;
      a6: pp 14    arn b9 ; each:=14;
      ca b16      pp 0 ; if t then each:=0;
      bs (b11)    pp 21 ; if x then each:=21;
      ca b27      pp 26 ; if pn then each:=26;
      gp a14      ca b27 ;
      pa a14      t 38 LZB ; if p then each:=38;
      ; new block:
b3:   bs 0        hv a7 ; if boobufVimage then goto same block;
      pa a9       pa a11 ; words left:=char left:=0;
      sy 64       arn b45 ; CR;
b4:   bs 1        hs b36 ; if addr print then write({dddd},a);
      srn e       ac b45 ; a:=a+1;
b5:  a7: arn a11   bs 0 ; same block:
      arn a9      sr a14 ; if boочar\char left>each
      hv a10      LT ; V-;boo char per line\words left<0
      ca 0        hv a10 ; then
      nt (b28)    pa r1 ;

```

```

      qq 0      ; hs b20 ; begin space(sp); goto same line end;
      a9: qq 0      ; hv a12 ;
b6: a10: it 0      ; pa a9 ; char left:=char per line;
b7: it 1      ; pa a11 ; words left:=words per line;
      a11: qq 0      ; sy 64 ; CR;
      can (b4)      ; hv a12 ; if -,addr print then goto same line;
      arn b47      ; hs b36 ; write({dddd},bi);
      bs (b3)      ; hv a12 ; if boobufVimage then goto same line;
      arn b46      ; hs b36 ; write({dddd},bj);
      a12: nt (a14) ; pa b28 ; same line: printed:=-each;
b8: can 0      ; hs b22 ; if kind=0 then get word kind 0
      bs (b8)      ; hs -26 ; else get word;
b9: hs 0      X ; ; write(form, word);
      can (b23)      ; hv a14 ; if first $\wedge$ g then
      arn b9      ; ; begin
      nc b14      ; hv a14 ; each:=printed+16; (i.e.printed+2);
      it (b28)      ; pa a14 ; sp:=2;
      qq (a14)      t16 ;
      pa b28      t-2 ; end;
      a14: nt 0      ; qq (a9) ; char left:=char left-each;
      qq (a11)      t-1 ; words left:=words left-1;
      srn e      ; ac b46 ; bj:=bj+1; bi:=bi+1;
      ac b47      ; sc b48 ; length:=length-1;
      pa b23      ; arn b48 ; first:=false;
      hv b0      ; LT ; if length<0 then goto next print list;
      arn b46      ; sr b44 ; if bj=blocklength then bj:=0;
      hv a7      ; NZ ; if bj $\neq$ 0 then goto same block
      grn b46      ; hv b3 ; else goto new block;
      e      ;
      ;
      b a26      ; FORM, NUMBER and TEXT
b10: bs (b11)      ; it a5 ; integer:
      pa a3      t a4 ; set layout;
      gr b49      ; IRC ; save marks
      gs a26      ; ann b49 ; save s
      tl -39      ; dl a6 ; R:=abs(number):1000000;
      hv a1      ; LZ ; if R=0 then goto small number;
b11: bs 0      ; hv a ;
      qq (a3)      t 1 ; if -,extra then layoutadr:=layoutadr+1;
      hv a1      ; ; goto small number;
      a: mt b49      ; gm b49 ; number:=abs(number)mod1000000;
      sy 0      ; NT ; if pos then space;
      qq (b28)      t1 NT ;

```

```

      qq (b28)      t1      ; count printed;
      qq a5      , hs b30    ; write({ddd ddd},R);
      pa b15      t 25      ; set leading zeros in print routine;
      sy 0        hv a2      ; goto second part;
a1:  can(b11)      hv a2      ; small number:
      qq 8        , hs b20    ; if extra then space(8);
      it 8        , qq (b28)  ; printed:=printed+8;
a2:  arm b49      ; second part:
a3:  qq 0        , hs b30    ; write(layout,number);
      pa b15      t 26      ; reset print routine
      hv a26      ; goto write mark;
a4:  qq 6.3+6.7+1.9+1.14+6.23; {-dddddd}
      qq 4.3+4.7+1.9+1.14+1.17+4.23; {-dddd0d}
a5:  qq 6.3+6.7+1.14+3.23+3.27; {ddd ddd}
a6:  1000000      ;
      ;
b12: bs (b11)      , it a9    ; fractional:
      pa a7        t a8      ; set layout;
      gs a26      IRC      ; store s, save marks
a7:  qq 0        , hs b31    ; write(layout,number);
      hv a26      ;
a8:  qq 4.3+1.7+1.9+3.13+1.14+3.17+1.19+5.23; {-d.ddd0-ddd}
a9:  qq 10.3+1.7+1.9+9.13+1.14+3.17+1.19+5.23+3.27+3.31;
      ; {-d.ddd ddd ddd0-ddd}
b13: bs (b11)      , it a9    ; real:
      pa a10      t a8      ; set layout;
      gr b49      IRC      ; save marks; RF:=R;
      arnfb49      , gs a26    ; save s;
a10: qq 0        , hs b32    ; write(layout,number);
      hv a26      ;
      ;
b14: it a21      , pa a15    ; group: set mask addr
      gs a26      IRC      ; store s;
      gr b49      , hvn a15    ; store R; goto take mask;
a13: pa a17      , arn a21    ; skip:=0;
a14: tk 10      , ck 0      ; next pos: mask:=mask shift 10;
a15: arn 0      t1 LZ      ; take mask: R:=mask;
      ga a16      V LB      ; set group length
      pa a17      , hv a26    ; exit
a16: bs 0        t 99      ; if grouplength>99 then begin
      ga a17      , hv a14    ; skip:=grouplength; goto next pos end;
      ca 0        , hv a14    ; if grouplength=0 then goto next pos
      gr a21      X          ; else store mask

```

```

      mkn a18      , mb a20      ;
      tk 2         , ar a19      ; construct layout
      gr b52       , pmn b49     ;
      bs (a17) X   ; if skip>0 then
a17:tk 0          , t -100      ; skip:=skip-100;
      cl (a16)     , gr b49     ; number:=numbershift(grouplength+skip)
      hs b30 X     ;
      qq b52       , hv a13     ; write(layout,numberpart);
a18:qq 154        ;
a19:qq 15.3+2.7+1.14+15.23;
a20:qq 15         ;
a21:Ob           ;
      10/10/10/10b ;
;
b16:  bs 0         , sy 60      ; text: if lastcase=UC then UC
      hv a24        , LZ       ; again: if R=0 then goto out
      cl 34         , ck -4     ; digit to RA
      nc 10         , ca 15     ; if RA=10VRA=15
a24:sy 58          , hr s1     ; cut: then begin LC; go back end;
      ca 63         , ar a25    ; if RA=63 then RA:=64;
      ca 60         , ga b16    ; if RA=60 then lastcase:=UC
      ca 58         , pa b16    ; if RA=58 then lastcase:=-,UC;
a25:ga r1         , xrn        ; writechar(RA); swap;
      sy 0          , hv 1b16   ; goto again;
;
a26:ps 0          , arn r      ; writemark:
      ca b19        , hr s1     ; if REGISTER PRINT/AM then go back;
      can s-b1      , hr s1     ; if blockhead then go back;
      sy 27         , V NRC     ;
      sy 49         , V NRB     ; write a b c or comma
      sy 51         , V LRC     ;
      sy 50         , LRB       ;
      hr s1         ; go back
;
e
;
b a8
; PRINT CONTROL REGISTERS:
b17: pmn b39      , hs b21     ; get first image track
      sy 53        , arn 9b33  ; writetext(<<e>);
      sy 56        , LB       ; if b marked then writetext(<<h>);
      qq a6        , hs b28    ; write(<p>,cell 9);
      sy 0         , sy 0      ; space(2);
      pmn b38      , hs b21     ; get last image track
      sy 39        , arn 36b33 ; writetext(<<p>);
      gp 40d14     , hs a5      ; store p; out(p);
      sy 18        , arn 31b33 ; writetext(<<s>);

```

```

hs a5 ; out(s);
sy 55 , arn 25b33 ; writetext({<g});
hs a5 ; out(group);
sy 19 , sy 34 ; writetext({<tk});
arn 26b33 , tk 10 ;
hs a5 ; out(tk);
arn 26b33 , ga a4 ;
sy 50 , sy 24 ; writetext({<by});
hs a1 ; out and bit(by);
sy 0 , sy 0 ;
sy 57 , sy 37 ; writetext({< in});
arn 27b33 , ga a4 ;
hs a1 ; out and bit(in);
hv b0 NKC ; if -,ka^-,kb then goto next print list
sy 34 ; writetext({<k});
sy 49 V NKB ; if -,kb then writetext({<a});
sy 51 V LKC ; if ka^kb then writetext({<c});
sy 50 LKB ; if -,ka^kb then writetext({<b});
hv b0 ; goto next print list;
a1: hs a5 ; procedure out and bit(no); integer no;
pa a3 t9 ; begin out(no);
a2: arn a4 , tk 1 ;
sy 1 V LT ; if bit then writetext({<1});
sy 16 ; else writetext({<0});
a3: bt 0 t-1 ;
a4: qq 0 , hh a2 ; end out and bit;
sy 0 , hr s1 ;
a5: hs b28 ; procedure out(no);
qq a6 , sy 0 ; begin write({p},no); space(2);
sy 0 , hr s1 ; end;
a6: qq 4.3+1.7+1.9+1.14+5.23; {p}
;
b18: gp 40d14 ; PRINT ARITHMETIC REGISTERS:
pmm b38 , hs b21 ; get last image track;
sy 60 , sy 41 ; writetext({<R});
arn 35b33 , pm 32b33 ;
sy 38 LB ; if overflow then writetext({<0});
tl 39 , pm 30b33 ;
sy 14 LO ; if LO then writetext({<1});
sy 2 LO ;
sy 58 , hs (b9) ; write(layout,R);
qq a8 , hs c23 ; writetext({< M});

```



```

b19:  arn 30b33 , hs (b9) ; write(layout,M);
      arn b9 ;
      ca b13 , hv r2 ;
      nc 1b13 , hv b0 ; if -,r then goto next print list
      qq a7 , hs c23 ; writetext({< RF});
      arn 32b33 , pm 30b33 ;
      grf b39 , ps b0-1 ;
      arn b39 , hv b13 ;
      a7: t RF;
      a8: t M;
      e
;
b20:  it (s) , pa r1 ; procedure space(i); integer i;
      bt 0 , t-1 ; for i:=i step -1 until 1 do
      sy 0 , hv r-1 ; writespace(1);
      hr s1 ;
;
b21:  tl 23 , tln -23 ; procedure get track;
      dln c11 , ar c11 ;
      tk 30 , ga b24 ;
      cl -10 , ga b25 ;
      vk (b24) , vk (b25) ;
      lk b33 , vk 0 ;
      hr s1 ;
;
      b a2 ; procedure get kind 0 word;
b22:  bs 0 , hv a2 ; begin if boobuf then goto bufferword;
b23:  bs 0 , hv a ; if first then goto set;
      arn b46 , pt b26 ;
      hv a1 , NZ ; if bj+0 then goto same track;
      a: pa a1 , t b33 ; set: set start addr
b24:  vk 0 ; select group
b25:  vk 0 , lk b33 ; read track to core
      is (b25) , t1 ; tk:=tk+1
      bs s-449 , t510 ; if tk=960
      pa b25 , it 1 ; then begin tk:=0; group:=group+1 end;
b26:  vk (b24) , it 0 ;
      a1: pmn 0 , t1 ; same track: get kind 0 word
      hr s1 ; go back
      a2: arn b50 , il 0 ; bufferword:
      srn e , ac b50 ; get buffer word
      pmn b33 , hv r-3 ; go back
      e ; end;
d d19=d19-960 ;
b38:  qq d19.29-d19.33+d16.39+25.39;
b39:  qq 26.23+d19.29-d19.33+d16.39;
b40:  qq 26.23+d19.29-d19.33+d16.39;

```

```

d d19=960d19 ;
b43: qq 4.3+4.7+1.14+4.23 ;
b44: 40 ;
;
;
;
;
;
b a21 ;
b64: ; ENTRY PAIR
<d35-2, pm r2 ud b53 ; rewind;
ah: gm b63 hs 40d14 ; set PAIR instr; get param track;
x pm r2 gm b63 ; set PAIR instr in PRINT
a: hs 40d14 ; next pair list: get param track
> hv b65 ; PAIR instr
pa b63 t b65 ; set return to PAIR instr in PRINT
<d41, pt b61 tc80-1c28; bufferbase:=1543;
pa b62 tc-1 ; corebase:=c-1;
> sy 1 hs b60 ; writetext({<1}); goto PRINT
b65: pm b50 gm 40b17 ; return after first run: mod area1:=mod area
pm c29-c28, gm 48b17 ; set addr for init med
pm b45 gm 41b17 ; a1:=a
pm b47 gm 44b17 ; b1:=b1
arn b46 gr 42b17 ; b1:=bj
tk 20 gt a15 ; start addr get kind 0 word 1
arn b8 ga a3 ; kind1:=kind
<d41, ca 2 it a19 ; bufferbase:=if kind=2 then 3086
pt b61 t a20 ; else 2190;
pa b62 t 53b17 ; corebase:=53b17;
> it (b22) pa a11 ; boobuf1:=boobuf;
arn b24 ga a13 ; group1:=group
arn b25 ga a14 ; track1:=track
pm b44 gm 43b17 ; blocklength1:=blocklength
pa a12 t1 ; first1:=true;
pa b63 t b66 ; set PAIR instr in PRINT
hs 40d14 ; get param track;
a1: qq 7.2+7.27 ; mask;
sy 2 hs b60 ; writetext({<2}); goto PRINT;
b66: pm c29-c28, gm 49b17 ; return after second run:
<d41, it (b8) bs 2 ; set addr for init med
hv r5 ; if kind<2 then goto loop;
arn b50 pm a1 ; if kind=kind1/unit=unit1 then
cm 40b17 hv r3 ; alarm(unit);
qq r1 hs c24 ;
tunit;
qq (40b17) tb17-b33; modify from buffer addr.
> a2: pm 48b17 pp c-1 ; loop: set bufferbase and corebase
a3: can 0 hs a11 ; get kind 0 word 1
bs (a3) ;
hs a18 hs c71 ; initialize get word, get word;

```

```

pa a12          IRC ; first1:=false;
gm b49          MRC ; store word 1
pm 49b17 , pp 53b17 ; set bufferbase and corebase;
can(b8) , hs b22 ; if kind=0 then get word kind 0
bs (b8)         ;
hs a18 , hs c71 ; initialize get word, get word;
a4: pa b23      IRC ; first:=false
gm 47b17 X     MRC ; store word
sr b49         IPC ;
hv a8          NZ ; if word=word1 then out pairs
qq (a5) t 1    LRB ;
qq (a5) t 2    LRA ;
qq (a5) t-1    LPB ;
qq (a5) t-2    LPA ;
a5: ncn 0      hs a8 ; if mark=mark1 then out pairs
a6: arn b48    ar e ; length:=length-1
hv a          IT ; if length<0 then goto next pair list
gr b48 , srn e ; bj:=bj+1; bj1:=bj1+1;
ac b46 , ac 42b17 ;
ac b47 , ac 44b17 ; bi:=bi+1; bi1:=bi1+1;
arn b46 , sr b44 ;
hv a7         NZ ; if bj=blocklength then
grn b46 , srn e ; begin bj:=0; a:=a+1 end;
ac b45        ;
a7: arn 42b17 , sr 43b17 ; if bj1=blocklength1 then
hv a2         NZ ; begin bj1:=0; a1:=a1+1; end;
grn 42b17 , srn e ;
ac 41b17 , hv a2 ; goto loop;
a8: arn b45 , sr 45b17 ; procedure out pairs;
ar 41b17 , sr 46b17 ; if a=lastprinted a
hv a9         LZ ; Va1=lastprinted a1 then
sy 64 , arn 41b17 ; begin CR;
gr 46b17 , hs b36 ; last printed a1:=a1; write({dddd},a1);
qq 41 , hs b20 ; space(41);
arn b45 , gr 45b17 ; last printed a:=a;
hs b36 , ; write({dddd},a);
a9: sy 64 , ; end;
arn 44b17 , hs a10 ; CR; write({dddd},bi1);

```

```

arn 42b17 , hs a10 ; write({dddd},bj1);
arn b9 , nc b27 ;
pa b28 V t-34 ;
pa b28 t-35 ; sp:=if p then -35 else -34;
pan b67 X t44b17 ; set bi addr in program print
arn b49 , hs (b9) ; write(word);
nt (b28) , pa r1 ; sp:=- (sp+printed);
qq 0 , hs b20 ; space(sp);
arn b47 , hs a10 ; write({dddd},bi);
arn b46 , hs a10 ; write({dddd},bj);
pan b67 X tb47 ; set addr bi in program print
arn 47b17 , hs (b9) ; write(word);
pa a5 , hv a6 ; go back
b36:a10:hs b30 ; end;
qq b43 , sy 59 ;
sy 0 , hr s1 ;

; procedure get kind 0 word 1;
a11:bs 0 , hv a17 ; begin if boobu1 then goto bufferword;
a12:bs 0 , hv r3 ; if first1 then goto set;
arn 42b17 , pt a15 ;
hv a16 NZ ; if bj+0 then goto same block;
pa a16 t b17 ; set: set start addr
a13:vk 0 ; select group1
a14:vk 0 , lk b17 ; read track to core
is (a14) t1 ; tk1:=tk1+1;
bs s-449 t510 ; if tk1=960 then
pa a14 it 1 ; begin tk1:=0; group1:=group1+1 end;
a15:vk (a13) it 0 ;
a16:pmm 0 t 1 ; same track: get kind 0 word 1
hr s1 ; go back
a17:arn 40b17 , il 0 ; bufferword:
srn e , ac 40b17 ; get bufferword
pmm b17 , hv r-3 ; go back
; end;
b37:a18:gp c68 , gp c73 ; initialize get word;
it p-1 , pa c66 ;
it p-2 , pa c67 ;
gm c65 , hh s ;
a19:3086 ;
a20:2190 ;
e ;

b a31 ; PROGRAM PRINT
b27: bs (b29) ;
sy (b29) t 510 ; LC;

```

```

      gr b49          MDC ; store word, set leftword and abs addr
a:   gs a22          IRC ; store s, halfword and f marks
      pp (b28)      t 6 ; p:=printed:=printed+6;
      pp p6          LOB ; if abs addr then p:=p+6;
      it p5          pa a9 ; char per halfword
a1:  qq 27          pp a29 ; p:=table start
      tk 20          ck 2 ;
      ga a3          ck -18 ;
a2:  gt a2          pm p0 ;
a3:  pin 0          X t-253 ; PA:=n; PB:=(); QA:=r; QB:=s; QC:=p;
a4:  tk 20          LTA ;
      tk 10          LTB ;
      cl 5          tk -15 ;
a5:  gt a5          ps 0 ; s:=letter no
      bs s-18        ps s-49 ; write instruction:
      bs s-9         ps s-25 ;
      sy s48         cln -20 ;
      hv a5          NZ ;
      sy 54          LRB ; writetext({<f});
      sy 37          V LPA ; writetext({<n});
a6:  sy 0          qq 0 ; if -.n then space(1);
      sy 0          NRB ; if -.f then space(1);
      sy 0          V NPB ; if -.() then space(1) else
arn a7          hs a11 ; writesymbol();
pm b49          arn a4 ; RA:=20;
hv a8          X NQC ;
ar -2          D LQB ; if svp then RA:=RA-2;
ar 21          D LQA ; if rvp then RA:=RA+21;
a7:  qq 520        hs a11 ; writetext( r , s or p);
      qq (b28)      t1 ; printed:=printed+1;
      hs b29          NQB ; if r then write({ddd},addrmod1024);
      qq a28          V NQB ; if r then skip next;
a8:  hs a30          ; write({ddd},addrpart);
      sy 0          V NPB ; if -.() then space(1) else
arn a20        hs a11 ; writesymbol();
hv a10          LQB ; if svp then goto no abs addr;

```

```

b67:   arn b47   , tk 30   ;
      ga r2     ;
      arn b49   IQB      ; QB:=abs addr
      ar 0      DVX LQC   ; if -,rv-,abs addr then goto no abs addr;
a9:   qq 0      , hv a10  ;
      arn a23   , hs a11  ; writesymbol([);
      hs a30    ; write({ddd},addrpart+bi);
      arn a25   , hs a11  ; writesymbol(1);
      qq (b28) , t2 IPB   ; printed:=printed+2;
a10:  pm a17   , arn b49  ; no abs addr:
      hv a21    NA      ; if right halfword then goto out;
      cm a6     NRA     ;
      hs a24    , pm a1  ;
      tk 10     , ps a   ;
      gr b49    X      MOB ; store right halfword;
      hh a14    X      NRA ; if fullword then goto full;
a11:  ga a13   , bs (b29) ; procedure write symbol;
      mt a20    , it 510  ; begin
a12:  sy (b29) , t-510 LT ; write case
a13:  sy 0      X      ; write symbol
a14:  hr s1     , tk 21  ; end;
      pp (b28) , t7 M    ; full: p:=printed:=printed+7;
      gp a9     , hs a15  ; write X
      qq 23     , hs a15  ; write V
      qq 21     , ps a16  ;
a15:  ud a12    NPB     ;
      pp (b28) , V t-1 NT ; procedure write X V D;
      sy (s1)   , IPB    ; begin
a16:  tk 1      , hh s1   ; end;
      qq 52     , pp p-4  ; p:=p-4;
      hh a19    LZ      ; if indicator part≠0 then
      hs a24    , tl -39  ; begin
      sy (b29) , t-510 NPB ; space(missing space);

```

```

tk 30      ga a18      ;      case
tln 3      ud 1a15     ;
a17:0/-1/0/-1      ;
tk 20      gt a18      ;      write indicator state
tl 32      ga a19      ;
ps a26     is a27      ;
a18:ud s0   ud s0      ;      end;
a19:ud s0   arn b49     ;
a20:qqf 521 gp b28      ;      printed:=p;
ca 0       hv a22      ;      if word=0 then goto out;
pm 19      DX          ;      space(missing space);
hs a24     hs a11      ;      writesymbol(t);
hs a30     ;      write({ddd}, increment);
a21:qq (b28) t1        ;      out: space;
a22:ps 0    hr s1      ;      restore s ; go back;
a23:qq 518  it 1       ;      procedure missing spaces;
a24:it (b28) bs (a9)   ;
sy 0       hh a23      ;
a25:qqf 519 hh s       ;
a26:pp p-1  pp p-1     ;      indicator letters:
sy 50      sy 34       ;      b, k
sy 49      sy 25       ;      a, z
sy 51      sy 38       ;      c, o
sy 37      sy 19       ;      n, t
sy 35      sy 39       ;      l, p
a27:sy 57   sy 40      ;      i, q
sy 36      sy 41      ;      m, r
a28:qq 4.3+4.23      ;      {ddd}
;      instruction letters: table start;
a29:qq 17.4+17.9+17.14+26.19+18.24+ 1.29+18.34+19.39
qq 14.4+ 1.9+14.14+19.19+ 3.24+ 1.29+ 3.34+19.39
qq 2.4+13.9+ 2.14+ 1.19+20.24+13.29+11.34+13.39
qq 12.4+13.9+11.14+ 4.19+12.24+ 4.29+11.34+14.39
qq 12.4+14.9+18.14+ 8.19+12.24+20.29+11.34+ 3.39
qq 12.4+ 3.9+18.14+ 7.19+ 1.24+ 7.29+20.34+ 7.39
qq 11.4+20.9+ 1.14+ 3.19+13.24+ 7.29+13.34+16.39
qq 18.4+24.9+ 9.14+ 7.19+19.24+16.29+16.34+16.39
qq 1.4+16.9+20.14+16.19+11.24+ 8.29+ 9.34+16.39
qq 19.4+ 9.9+20.14+ 9.19+13.24+ 3.29+20.34+ 2.39
qq 19.4+14.9+20.14+14.19+16.24+ 7.29+ 3.34+14.39
qq 12.4+ 9.9+19.14+21.19+ 7.24+ 7.29+ 3.34+ 7.39
qq 3.4+16.9+19.14+ 2.19+19.24+ 8.29+25.34+22.39
qq 11.4+12.9+11.14+19.19+11.24+ 7.29+11.34+22.39
qq 22.4+ 8.9+10.14+26.19+25.24+19.29+25.34+12.39
qq 8.4+ 8.9+19.14+ 7.19+12.24+26.29+ 4.34+21.39
a30:sr 983 D          ;
it b29          NT    ;      if number>983 then
pa a31          t b28  ;      write({ddd}, number mod 1024)
ar 983 D        ;      else write({ddd}, number);
a31:hs 0        ;
qq a28          hr s1  ;
e              ;

```

d b28=i, b29=1b28, b30=1b29, b31=1b30, b32=1b31, b15=45b28

d b45=123i, b46=1b45, b47=1b46, b48=1b47, b49=1b48, b50=1b49, b52=1b50

[NUMBER PRINT ROUTINE

page 1]

```

b a51
;
a42: qq 0 ; ck 0 ; entry address part 0<x<1024
a50: qq 570 ; tl -30 ; entry address part -512<x<511
pm 28 DV ; entry integer
pm -11 D ; entry fractional
pa ra11 X t485 ; entry real; numberpart:=true;
gs ra12 ; gp ra13 ; save p; save s;
ga ra14 ; gm ra23 ; store exp2; store numberpart;
pm (s) ; arn s ;
pm (s1) NA ; get layout address
a14: psn 0 X ; s:=exp2;
ps s11 ; cl -20 ; s:=exp2+11;
tk 14 ; ga ra17 ; unpack layout: b
tk 10 ; pa ra16 ; exp10:=0;
a40: it 0 ; pa ra4 ; expprinting:
pp 256 ; ck -6 ; bE:=256;
ga ra18 ; tk 10 ; h
ck -8 ; ga ra19 ; f1
tk 10 ; ck -6 ;
ga ra20 ; tk 11 ; d
tk -20 ; gt ra21 ; -n
tk 20 ; ck -1 ;
ga ra22 ; tk -6 ; bE+f2
ca 1 ; pp 10 ; if bE=1 then minexp:=10;
ca 2 ; pp 100 ; if bE=2 then minexp:=100;
tln 34 ; ar ra43 ; group picture
gr ra8 ; snn ra23 ; R:=-abs(numberpart);
pa ra4 V t-15 LZ ; if numberpart=0 then H:=-15
a49: nk ra46 XV ; reconversion: else x:=numberpart*x2/exp2;
gr ra26 ; hv ra31 ; goto if numberpart=0 then L3
bs s-11 ; hv ra47 ; else if s>0 then conversion1
; else conversion2;
a34: hv ra48 ; sr ra26 ; round x2: R:=rounded x;
a17: pa 0 XVD t11 NO ; if -,overflow then b:=11
mt ra7 ; hv ra49 ; else goto reconversion;
bs (ra4) ; ntn (ra4) ; if H>0 then begin R:=0;
a21: qq (ra18) ; h:=h-H end else h:=h-n;
a19: pp 0 ; gt ra13 ; p:=f1; a13incr:=R;
bs p509 ; hv ra36 ; if p+3 then goto count h
a11: arn 485 D t-485 ; exppart:=,numberpart; numberpart:=false;
hs ra5 ; LT ; if exppart then write 10
bs (ra24) ; arn ra23 ; if b1>0/x<0 then

```


[NUMBER PRINT ROUTINE

page 2]

```

arn -480 DV      NT ; R:=
arn 32 DV        ; else R:=+
bs p510 , ck 10  ; if R<0^p<2 then R=small
hs ra37 , pp 3   ; write sign
a36:             ;
a18: bt 0        t-1 ; count h:
hsn ra2 , hv ra36 ; write space before digits
bs p509 , hv ra11 ;
a13: pp 0        ncn 0 ; restore p;
hsn ra1 , qq     ;
bt (ra4) , t-1    ; count H
hsn ra , hv r-1   ; write digits before point
arn ra38 , bs (ra20) ; if d>0 then
hs ra5 , it -1    ; write point;
a20: bt 0        hh ra39 ; count d, write decimals
a12: ps 0        xrn     ; restore s; M:=0;
bs (ra24) , pm ra16 ; if b2>0 then M:=exp10;
can (ra22) , hr s1 ; EXIT
gm ra23 , srn ra9 ;
pm (ra22) DX      ;
ps 9 , hh ra40 ; goto expprinting;
a37: ca p , hh s ; if p=0^Raddr=0 then go back
gr ra26 , arn ra9 ; sign not counted in group
ac ra8 , arn ra26 ;
a5: bs (ra) , hv ra1 ; if b1>0 then write digit
mb ra44 , ga ra6 ;
a39: hvn ra10 , arn ra3 ; else if digit=0 then write digit
a4: it 0 , t1 ; count H else write 0
bs 0 , hvn ra5 ; if H<0 then write 0
a: bt 0 , t-1 ; count b1
mln ra3 , tk 30 ; next digit in R
a1: ar 16 D LZ ; zero instead of space
a2: ga ra6 , bs (ra50) ; if actual case=upper then
mt ra7 , it 510 ; R:=-R;
sy (ra50) , t-510 LT ; write case;
a10: qq (ra42) , t1 ; count printed;
a51: arn ra8 , sr ra9 ; actual group:=actual group-1;
gr ra8 , nc -273 ; if -;group full then write out
a6: sy 0 , hhn s ; else if out>58 then begin
tk 4 , it 58 ; actual group:=next group;
bs (ra6) , hv r1a51 ; write out end

```

[NUMBER PRINT ROUTINE

page 3]

```

a44: sy -256 , ud ra10 ; else begin writespace; count printed;
; actual group:=next group-1; write out end;
a41: hh ra51 , it 1 ; conversion:
a28: qq (ra4) , t-1 ; begin comment (count H)
a25: ps s-3 , nk r1 ; by multiplication by 213/10 or 10/214
ps s0 , gr ra26 ; x is converted to form
pm ra26 , bs (ra46) ; x=x2x1014H where x>x2>.1
a47: mkn ra27 , hh ra41 ; conversion 1:
a46: tk s , gr ra26 ;
a48: ps s7 , ar ra29 ; conversion 2:
pm -1 DV LT ;
mkn ra30 , hv ra28 ; end conversion;
ps (ra4) , can (ra22) ; s:=H; if bE=0^f2=0 then begin
bs s-15 , it 64 ; if s>15 then bE=1 end;
a22: ca 0 , hv ra31 ; if bE^f2=0 then goto L3;
arn ra17 , sr ra20 ; R:=b-d;
ga ra32 , sr ra18 ; a32:=b-d; Raddr:=b-d-h-1;
a7: mb -1 DX LT ; M:=if b-d-h-1<0 then b-d-h-1 else -1;
a38: xr 315 , it (ra4) ; L1: R:=M;
a32: bs 0 , hs ra33 ; if b-d>H then change exp10
mt ra7 , it (ra18) ; L2: R:=-R;
bs (ra4) , hs ra33 ; if H>b then change exp10
a31: arn ra20 , ga ra ; L3: b1:=d;
a45: ar ra4 , ga ra24 ; L4: b2:=H+d;
a24: bs 0 , ga ra ; if b2>0 then b1:=b2;
it (ra17) , bs (ra24) ; if b<H-d then
arn ra17 , hh ra45 ; b2:=b; goto L4;
arn 256 D NT ; if b2>0 then R:=.5;
a35: ps (ra24) , pm ra21 ; rounding: s:=b2;
bs s511 , hh ra34 ; if s<0 then goto round x2;
a16: xr p0 , mln ra29 ; R:=Rx.1;
ps s-1 , hh ra35 ; s:=s-1; goto rounding;
; change exp10:
a33: ac ra16 , bs (ra16) ; exp10:=exp10+R; if exp10>minexp then
sc ra4 , hh s-1 ; begin H:=H-R; goto L1VL2 end;
sc ra16 , hv ra31 ; else exp10:=exp10-R; goto L3;
;
a3: 10 ;
a9: qq 1 ;
a27: can s409 , cm (r-410) ; 0.8
a29: vy p51 , mln (204) ; 0.1
a30: qq 320 ; 10/16
a43: qq -17.5+1.25-1.39 ;
d a8=i, a26= 1i, a23=2i ;
e ;

```

```

i=39i, d=k-d1      ;
b k=d42, i=0, a10   ;
a1=d19-960          ; a1=image group
a2=40d14            ; a2=first cell in core
a3=b53, a7=b64      ; a3=entry print, a7=entry pair
a=d, <d35, a=2>      ; a=number of blocks
<d39,               ; if aux only then
i=d2                ; begin
    hs 1            ;
    hv a4           ;
<d36,               ; if aux reserved then
    tres;           ; res, no of blocks
    qqf a.39        ;
x<d39,              ; else
    tset;           ; set,
    qqf d35.39      ; aux kind,
    qqf a.39        ; no of blocks,
[STOP, SUM] i base print, pair
s                   ; typein
[STOP, CLEAR]      ;
><d39,              ;
    qq 39;          ; concat pid 0,
    qq 57;          ;
    qq 52;          ;
    qqf             ;
    tprint;         ; print, spec<
    qqf d.9+a3.19+a2.29 ;
    tpair;          ; pair, spec<
    qqf d.9+a7.19+a2.29 ;
    qqf,            ;
a4: hs 1            ;
    hv a5           ;
    tmove;          ; move, b loadplace, print<
    qq 50;          ;
    qqf d.23+d1.39+a1.29-a1.33;
    tprint;         ;
    qqf,            ;
a5: hs 1            ;
    hv a6           ;
    tsetsum;        ;
    tprint;         ;
    qqf,            ;
d2=i               ;
a6: hsf 2           ; end
xi=d48             ; else load to primitive catalog
    qq d35.2+11.7+d36.5+d.23+d1.39;
    qq,             ;
    tprint;         ;
    qq d.9+a3.19+a2.29 ;
    tpair;          ;
    qq d.9+a7.19+a2.29 ;
d48=i              ;
    qqf             ;
> d1=d+d1          ;
e                   ; end image load
e                   ; end print, pair
[STOP, SUM] i print, pair
s                   ;

```

[Call: run, <name> <

<name> ::= <name of drum area>|<empty>

This aux program will execute a translated ALGOL program described by <name>. If no parameter is present the ALGOL program will be taken from the work area]

[Here follows STOPCODE,CLEARCODE]

```

b k = d1, i = 238, a10, b6 ;
d a=15 ;
b1: pa 8 t 1 ; ENTRY: set no init of cells[0:9];
a1: pp d13-1 , hs c52 ; p := start parameterlist - 1; GET PARAM;
[1] qq e5 , hs c24 ; if parameter ≠ name then alarm(<<param>);
a4h: hh a3 , mb a5 ; if parameter = end list then goto NO PARAM;
nc 0 , hh 1a1 ; NAME:
tl -16 , gr 261a ; if kind ≠ 0 then alarm(<<param>);
tln 16 , gr 264a ; tracks occupied in top of free := bits(8,23,areaw);
ar a8 , gr 263a ; first track program := bits(24,39,areaword);
arn 264a , ar 261a ; execution end track := first track program + 1;
sr a8 , sr b3 ; if first track program < last track image
qq V LT ; ^ last track program > first track image
arn b4 , sr 264a ; then set half inhibit
grn -1 V LT ; else set no inhibit;
acn -1 MA ;
vy d43 , arn 2c ; select help alarm output unit;
tl -4 , nc 7 ; TEST BITS: if reserved bit of areaword = 0
nc 5 , grn 261a ; ^ special bit of areaword = 0 then
pm 264a , hsn c3 ; tracks occupied in top of free := 0;
lk 265a , vk (c4) ; to core(first track program,265e4); wait drum;
arn a6 , sr 267a ; if instruction[2] of program
pa 1a1 t a7 ; ≠ lk p , hh r-2
hh 1a1 NZ ; then alarm(<<not present>);
hh 1a1 NA ;
hh 1a1 LB ; goto instruction[0] of program;
a3h: hv 265a , hsn c2 ; NO PARAM: get free;
lk d14 , vk (c4) ; to core(first catalog track,d14); wait drum;
arn 3d14 , gr 2c ; R := areaword := work; goto NAME;
hh a4 ;
;
a5: qq -1.39-31.7 ; mask;
a6: lk p , hh r-2 ;
a7: tnot present; ;
a8: qq 1.39 ; 1
d b6=d19-960, b5=25d16 ; group no for image; last track image;
b3: qq d16.39+b6.29-b6.33; first track image;
b4: qq b5.39+b6.29-b6.33; last track image;
qq [fill] ;
d b=259a ;
b: ;
[259e4] qq [lower limit in buffer.39] ; EXECUTION PARAMETERS:
[260e4] qq [initial last used in buffer] 4096.39 ;
[261e4] qq [tracks occupied in top of free.39] ;
[262e4] qq [abs track look up] c64-1 ;
[263e4] [execution end track.39] ;
[264e4] [first track program.39] ; not loaded;
<j,i load error, run ; not loaded;

```

>

```

i=i+39, d=k-d1          ;
b k=d42,i=0,a10         ;
a1=d19-960              ; a1=group no for image
a=d,<d35,a=1>            ; a=no of blocks
                           ....
<d39                     ; if aux only then
i=d2,hs1                 ; begin
    hv a5                 ; (if aux reserved then
<d36,tres;               ;
    qqf a.39              ; res,no of blocks,
x<d39                     ; else
    tset;                 ; set,aux kind,no of blocks,
    qqf d35.39            ; type in)
    qqf a.39              ;
[after i follows STOP,SUM and a sum character]
ia base,run
s
[STOP,CLEAR]
x<d39                     ;
    qq 39,                ; concat pid 0,
    qq 57,                ;
    qq 52,                ;
    qqf                   ;
    trun;                 ; run,spec<
    qqf d.9+b1.19+b1.29   ;
    qqf,                  ;

a5: hs 1                  ;
    hv a6                  ;
    tmove;                 ; move,b loadplace,run<
    qq 50,                 ;
    qqf d.23+d1.39+a1.29-a1.33 ;
    trun;                 ;
    qqf,                  ;

a6: hs 1                  ;
    hv a7                  ;
    tsetsum;               ;
    trun;                 ;
    qqf,                  ;
d2=i                      ; end else
a7: hsf 2                 ;
x                          ;
i=d48                     ;
    qq d35.2+11.7+d36.5+d.23+d1.39,, load to primitive catalog;
    qq,                   ;
    trun;                 ;
    qq d.9+b1.19+b1.29    ;
d48=i,qqf                 ;
>                          ;
d1=d+d1 ;                 ;
e                          ; end image load
e                          ; end run
[after i follows STOP,SUM and a sumcharacter]
ia run
s
s

```

[27.11.67

(3)

Slip, page 1]

[STOP, CLEAR]

<-d55+1, i version

s
>

[Navne globale til SLIP før SLIP	efter SLIP
c	plads for search i HJÆLP	
c2	indhop i search	
c4	vk kanal i search, indhop til pak op og vælg	
c5	vk gruppe i search	
c7	lk-ordre til læs katalog	
c18	bruges ved indhop i search	
c30		første kanal i SLIP
c31	.	antal kanaler i SLIP
c32	.	c2-c
c33	.	c4-c
c34		c5-c
c35	.	c7-c
c36		2c18
c37	.	2c
c38		indhop i SLIP
c39	.	antal kanaler som hentes med HELP
c63	init medium kanal	
c70	baandfejl	
c71	get word from buffer	
c77	return if sum	
d1	første fri kanal	første fri kanal efter SLIP
d2	første fri celle for init	samme efter slip
d9	første kanal i HJÆLP	
d10	ferritlagerbillede-320	
d11	parameterkanal	
d12	plads i HJÆLP til kanal med adressetrykprogram	
d14	kanalplads for inputstreng i HJÆLP	
d15	længde af texttrykprogram	
d17	by-værdi for l	
d19	den gruppe SLIP læser til	
d20	plads for SLIP i FL	
d21	første katalogkanal	
d22	antal kanaler/320	
d24	i adressetrykprogram	
d25	-	
d26	-	
d27	-	
d28	-	
d29	-	
d30	-	
d31	-	
d32	work in free	
d35	medium kind	
d36	reserved	
d37	texttryk kanal-d9 d37-1 er search kanal	
d39	input mode	
d40	adressetrykprogram i HJÆLP	
d47	return to init help	
d48	init katalog	
d50	image gruppe under indlæsning af slip]	

m

<d35, c30=d1, x c30=6d1> ; gør plads til 6 kanaler hvis tromle
c32=c2-c, c33=c4-c, c34=c5-c, c35=c7-c, c36=2c18, c37=2c

b k=c30, i=d20, a24, b21, c18, d7, e123

e64=d9, e88=d10, e103=d17, e98=d19

d=-27, d1=d-2, d3=d1-7, e38=d3-40, e84=e38-d15, e123=40d12-d40

e95=e84-e123, e39=e95-40, e50=e39-41

b a [navne i adressetrykprogram]

a=d40-e95, e81=d24-a, e80=d25-a, e77=d26-a, e79=d27-a, e78=d28-a, e32=d29-a

e33=d30-a, e83=d31-a

e

[navne paa variable]

a14=3d, e66=a14, a15=1a14, e47=1a15, e45=1e47, d5=1e45, a17=8d5, e30=a17

e48=1a17, a23=1e48

d6: qq 63.25

qq 1.1+3.16+7.24+5.32

qq 1.26+1023.39

d7: e8=d7-1

[CP-code table]

; Differens tabel

1.sym 2.sym værdi 1.sym 2.sym værdi

qq 512.9			-55.9-34.15	
qq 512.9	+55.9+34.15	+57.27+35.33+44.39	-57.9-19.15+37.21	; il, it
qq -57.27-35.33-44.39+57.9+19.15-37.21	+57.27+18.33+36.39	-56.9-56.15+60.21		; is, hh
qq -57.27-18.33-36.39+56.9+56.15-60.21	+56.27+41.33+17.39	-56.9-34.15+34.21		; hr, hk
qq -56.27-41.33-17.39+56.9+34.15-34.21	+56.27+21.33+56.39	-56.9-18.15+50.21		; hv, hs
qq -56.27-21.33-56.39+56.9+18.15-50.21	+55.27+57.33+29.39	-55.9-55.15+46.21		; gi, gg
qq -55.27-57.33-29.39+55.9+55.15-46.21	+55.27+51.33+47.39	-55.9-49.15+22.21		; gc, ga
qq -55.27-51.33-47.39+55.9+49.15-22.21	+55.27+41.33+21.39	-55.9-39.15+42.21		; gr, gp
qq -55.27-41.33-21.39+55.9+39.15-42.21	+55.27+36.33+26.39	-55.9-34.15+54.21		; gm, gk
qq	55.9+34.15	-39.9-57.15		
qq 512.9	+39.9+57.15	+55.27+19.33+23.39	-55.9-18.15+61.21	; gt, gs
qq -55.27-19.33-23.39+55.9+18.15-61.21	+52.27+35.33+14.39	-52.9-34.15+13.21		; dl, dk
qq -52.27-35.33-14.39+52.9+34.15-13.21	+51.27+49.33+25.39	-51.9-36.15+38.21		; ca, cm
qq -51.27-49.33-25.39+51.9+36.15-38.21	+51.27+35.33+20.39	-51.9-34.15+19.21		; cl, ck
qq -51.27-35.33-20.39+51.9+34.15-19.21	+50.27+19.33+39.39	-50.9-18.15+49.21		; bt, bs
qq -50.27-19.33-39.39+50.9+18.15-49.21	+49.27+51.33+ 6.39	-49.9-50.15+ 9.21		; ac, ab
qq -49.27-51.33- 6.39+49.9+50.15- 9.21	+49.27+41.33+ 2.39	-49.9-37.15+ 4.21		; ar, an
qq -49.27-41.33- 2.39+49.9+37.15- 4.21	+40.27+40.33+ 0.39	-39.9-57.15+35.21		; qq, pi
qq	39.9+57.15	-35.9-34.15		
qq 512.9	+35.9+34.15	+39.27+51.33+48.39	-39.9-49.15+32.21	; pc, pa
qq -39.27-51.33-48.39+39.9+49.15-32.21	+39.27+39.33+31.39	-39.9-36.15+27.21		; pp, pm
qq -39.27-39.33-31.39+39.9+36.15-27.21	+39.27+19.33+33.39	-39.9-18.15+30.21		; pt, ps
qq -39.27-19.33-33.39+39.9+18.15-30.21	+37.27+51.33+43.39	-37.9-35.15+16.21		; nc, nl
qq -37.27-51.33-43.39+37.9+35.15-16.21	+37.27+34.33+15.39	-37.9-19.15+41.21		; nk, nt
qq -37.27-34.33-15.39+37.9+19.15-41.21	+37.27+18.33+40.39	-36.9-50.15+ 8.21		; ns, mb
qq -37.27-18.33-40.39+36.9+50.15- 8.21	+36.27+35.33+12.39	-36.9-34.15+11.21		; ml, mk
qq -36.27-35.33-12.39+36.9+34.15-11.21	+36.27+19.33+10.39	-35.9-34.15+52.21		; mt, lk
qq	35.9+34.15			
qq 512.9		+35.27+24.33+59.39	-25.9-40.15+ 1.21	; ly, zq
qq -35.27-24.33-59.39+25.9+40.15- 1.21	+25.27+35.33+62.39	-25.9-33.15+57.21		; zl, zj
qq -25.27-35.33-62.39+25.9+33.15-57.21	+23.27+41.33+28.39	-21.9-34.15+55.21		; xr, vk
qq -23.27-41.33-28.39+21.9+34.15-55.21	+21.27+24.33+51.39	-20.9-52.15+63.21		; vy, ud
qq -21.27-24.33-51.39+20.9+52.15-63.21	+20.27+18.33+45.39	-19.9-35.15+18.21		; us, tl
qq -20.27-18.33-45.39+19.9+35.15-18.21	+19.27+34.33+24.39	-18.9-51.15+ 7.21		; tk, sc
qq -19.27-34.33-24.39+18.9+51.15- 7.21	+18.27+41.33+ 3.39	-18.9-37.15+ 5.21		; sr, sn
qq -18.27-41.33- 3.39+18.9+37.15- 5.21	+18.27+34.33+53.39	-18.9-24.15+58.21		; sk, sy
qq -18.27-34.33-53.39+18.9+24.15-58.21				

```

e19: ar      D      ; r
e62: d2: pa  D  t3  ; m
[1e62] 40      ;
[2e62] 153     ;
[3d2]   qq 1.3-1.15 ;
[4d2]   qq e39-1   ;
[5d2]   qq         ;
[6d2]   qq 1.39    ;

```

[ordrelæsning]

```

b a12, b8      ;
e4: gk         VD      ; nyt ord med mulighed for adr tryk.
[1e4]pa e4     ; nyt ord uden mulighed for adr tryk.
      ps 6e4    , pi    ;
      pi (e4)   IK t-20 ;
      grm d     XM      ;
      pi       , grm e27 ; forbered ny linie
[6e4]hh e20    ,      ; første symbol
e29: gr 1d3    , hv p+a1 ; gem 1 op. hopordbog
e7:a1:hv e28   , hv e9   ; 0-9 tal, adr
      hv a2    , hv e9   ; a-eik, op ell lab, adr
      hv e28   , hv e9   ; +- tal, adr
      hv e28   , hv c1   ; . x tal, fejl
      hv a3    , hv b1   ; , /CR terminator1, 2
      hv a4    , hv b2   ; AXS)...op, mod
      hv a4    , hv b3   ; mprst(op, start adr,
      hv a4    , hv c1   ; =:h op fejl
a2: it (4d3)   , pa e5   ; pos:= y
      hs e20    ;
      bsp +511  , hh a11  ; test g = 0, label m index
      bs p-6    , hvn a12 ; test g = 7 label
e93: hv a5     , sy 64    ; op, skriv vr efter s
      pa e83    , hv e4   ; klar til kanalnr. udskrift

a3: ca 64     , hv e4    ; blank CR
b1: ca 64     , hv a6    ; linie slut
      hv c1    LTB      ; test om hhao tilladt
      bs s-2   , hv c1   ; test om for mange adr
      pi 32    t -33     ; sæt hhao
      hs e20   , qq 1    ; læs 1 op
      gr1 d3   , ca 64   ; test for linie slut
a6: ps e4-1   , HV e43   ; gem 1 op, helord lagres
a4: hs e20    ; læs resten af op
a5: ck -6     , ac 1d3   ; op karakterer til 1d3
      hs e36   ; hent op-tabel
      arn 1d3  , pm e19   ; R:= op karakterer, M:= relativmærke
      pa a7    V t e8    ; op-opslag
      it 9     NO       ;
a7: ar        t 1       ;
      hv a7-1   NT      ;
      gm e27    , pm 3d2  ; sæt relativmærke, måske i M
      cm 5d2    , hh a8   ; hop hvis ikke venstre tabel

```



```

a9: ck -4 , mb d6 ; operation paa plads
a10: ck -10 LPA ; evt hhao
      ab d ; til helord
      gr d MA ;
      hsn e20 X ; læs 1 adrcif
a8h: hhp +a1 , ck 18 ; hopordbog
      cm1 d3 , hv c1 ; forbudt operation
      hv a9 ; operation OK
b2: pm 1d6 , it (4d3) ;
      bs 44 , hvn b4 ; , -, SnfF)
      ca 54 , itn -17 ; ff
b5: pi 80 LZ t-65 ; Fmærke, forbyd hhao ell blind
      tln(4d3) v ; skift
b4: hv b5 NPA ; hop til OK hvis -, hhao
      hv c1 LZ ; fejl hvis 0
      mb 2d6 , hv a10 ; accmuler bit

b3: ca 19 , hh b6 ; t
      bs s-1 , hv c1 ; , fejl(tællede1)
      ca 8 , hv b7 ; (
b8: arm 4d3 , ca 4 ; , sæt rel mærke
      hv c1 ; fejl hvis ( eller t
      ga e27 , HV e10 ; læs adr m. e20
b7: hs e20 ;
      pt e27 t 4 ; sæt (
      bsp +505 t 510 ;
      hv b8 ; g = 6
b6h:e68:hve9 , grn e27 ; læs adr, slet rel mærke
      bs s+510 , ps s+2 ; , sæt tællede1læsn
[navn]a11h:hve10,hs e24 ; læs heltal, ciffer læst
a12: bs p505 , hv c1 ; g ≠ 7
      gr d , it (e5) ;
      bs 3 , hv e53 ; i eller k
      pa e49 t 1 ;
      arm 4d3 , ca 47 ; ,h
      pa e49 , hs e20 ;
      arm 4d3 , ga e52 ;
      ca 48 , hhr +1 ; :
      hv e12 , arm d5 ;
      ck 30 , hs e51 ;
      hv 1e4 ;

e
b a20 [adresselæsning]
e17: arm d3 , ca 64 ;
      hh s ;
e10: hsn e20 X ;
e9: pa e25 , gm 1d1 ; fortegn, adr:= 0
      gm d1 , it s+509 ; term s = 0
a11: pt a1 ; def, ciffertæller
      hsn e22 X ; læs et led
      bsp +509 , hvp +a2 ; p = 0 ∨ 1 ∨ 2
a12: bss +509 , hvn a3 ; s = 0 ∨ 1 ∨ 2 udhop ordre
      bss +507 , hv c1 ; s = 3 ∨ 4 for mange adresser
      arm 1d1 , it 512 ;
      bsp +508 , hv c1 ; fejl hvis terminator ≠ , ∨ / ∨ CR
e15:a18:hvs+1 , qq 1 ; bruges som konstant
a2: pp (2d3) , hv a5 ; , heltal
      pp (e5) , hv a6 ; , navn
e25: [fortegn] , hv c1 ; , fejl(++

```

```

a6:  bs p-2      , hv a7      ; , a - e
      hv c1      , NZ      ; cifre efter i eller k
      arn p+d5   , hv a8      ; hent i eller k

a7:  hs e16      ; opslag
      hv a9      , NRA      ; udefnavn
      tl -30     , pp (e27) ; , rel msk
      cap -2     , sr d5     ; , -i
a8:  ar d1      , pp (2d3)   ; + term,
a14: bsp +508   t 510       ;
      bs (4d3)   , hh a10    ; , .
a15h:tk 30     , mt e25     ; adr paa plads, fortegn
      ac 1d1     , it 510    ; adresse
a13: bsnp+510   , hv c1     ; fejl hvis navn følger term
      gr d1      , hv a11    ; term = heltal ell 0
      tk 30      , mt e25    ; term var heltal
      ac 1d1     , hv a12    ; . udhop
a5:  bsp +509   , hvp +a13   ; for heltal: switch paa termin.
a10h:hv a14    , gr d1      ; , efter
      hsn e26    X          ; læs rent heltal
      tk 20      , gt r+1    ;
      arn d1     , ns        ;
      cls +39    , hh a15    ; skift term, akkumuler

```

```

[undef navn]
a9:  pp (2d3)   , nt (e25)   ;
      bs 1      , it (e47)   ;
a1h: bs (e5)    , it        ;
      bs 1      , hv c2      ; forbudt udef. navn
      bsp +508   , hv c2      ;
      bss        , ck 10     ;
      ga 1d1     , tl -30    ;
      arn d5     , tl 30     ; i
      bss        , ck -10    ;
      ck 20      , LRB       ;
      GR (e5)    , hvn a12   ; sæt i ordbog, udhop ordre]
a3:e27:arn D    ; relativmærke ind
      ck 20      , ab 1d1    ; , ; , adresse
      bss        , cl -10    ; , hhao
      pss +2     , pt d      ; tæl s, tælledel = 0
      ab d       , gr d      ; helord, helord
      grn e27    , pm d3     ; relativmærke, symbol
      hhn p+e7   X          ; hopordbog b

```

```

[opslag]
e16: tk 30      , ck -1      ;
      ar a18     , ga a16    ;
      gt a17     , it d5     ;
e5:  arn        , ga r       ;
      ca 512     , hv c3     ; ikke erklæret
      gt r       , it 0      ;
e18h:a16: bs    , hv e5     ; hop index > længde
      gr e30     , it (a16)  ;
a17h:arn(e5)    , bs        ;
      ck 20      V IRA      ;
      pi 5       V IQA -264  ;
      pi 8       V IQB -266  ;
      pi 2       LB -3       ;
      ck 20      , hr s+1    ;

```

```

[h = def adr eller = def adr]
e13: hsn e20      X      ; læs ciffer
e11: pa e52      , arm 4d3 ; redef. lovlig
     ca 49      , hv e10  ; læs defineret adresse
     hv c1      ;

```

e

```

b a20
e22: arm 4d3      , pp (2d3) ; læs et adresseled
     bsp -2      , hr s+1   ; p ≥ 3 udhop
a18: gp 3d3      , ga 5d3   ;
     bs p        , hv a16   ; p ≠ 0
     hs e24      ; p = 0 læs heltal
     pp (3d3)    , hr s+1   ;

```

```

a16: bs p-1      , hv a17   ; p = 2 fortegn
     it (4d3)    , pa e5     ; p = 1 navn
     hs e26      ;
     pp (3d3)    , hr s+1   ;

```

```

a17: it (e23)    , pa e25   ;
     hs e20      ;
     bs p-1      , hr s+1   ;
     hv a18      ;

```

```

e24: arm 4d3      , ck -30   ;
     ml e95+e123-2 ;
e26: hs e20      ;
     bsp +511    , hv e24   ;
     hrn s+1     X      ;

```

[Den centrale læsesekvens]

[læs et symbol]

```

e14: arm d3      , ca 64    ; som e20 men udhop for CR
e96: qq [checksum], hhs     ;
e20: pp          , lyn d3    ; comment situation
[1e20]ga d3      , tk -1    ; lige-ulige
     ga a2      ;
     bs (a2)    NT t 32     ;
     hh a3      ;
     gt a1      , it e1     ;
a1:a2: arm       , it       ;
     bs        , ck 10     ; skift for ulige
e23: ck -10[case], tk -7    ; case tilføjes
     ga 2d3     , sr 2d3    ; g
     ck 7       , ga 4d3    ; y
     nc 124     , hh a20    ; hop ikke sum code el. clear code
a20: hvn e41     , is (d3)   ;
     it s1      , qq (e96)  ; summer e96:= e96 + char - 63
e2:  bs        , hv a4      ; evt fejludskrift
     nc p92     , hv a12    ;
     pa e32     t 511      ;
     sy 29      , an(e83)   ;
     arm 1d5    , hs e81    ;
     arm d5     , hs e81    ;
     ps 6e4     , sy 62     ;
a15: qqn 64     X          ; bruges i input fra tromle
a12: ca 126     , hh e20    ; blind ogsaa i strenge
     bsp -19    , hv a5     ; , comment eller streng

```

```

a11: pp (2d3) , it 100 ;
      bs (4d3) , hv a5 ; internt symbol
a10: arn d3 , hr s+1 ; symbol, udhop
a5: ca 125 , hh a6 ; caseshift
      bs p-250 , hv pa8 ; streng eller comment
      ca 123 , hv a9 ;
e91: ca 122 , pp 300 ; ; skip til VR
      ca 121 , pp 301 ; [ skip til ]
      ca p102 , ppn 302 ; x skip til x eller >
      ca p-148 , hv s1 ; skip til VR efter fejl
a19: a8=a19-300, e82=a10-a8
      ca p-200 , hv a11 ; comment slut VR eller skip til ,/:
      ca 120 , hv c5 ; fejl
      ca p-200 , hv e20 ; comment slut ] eller >
a3h: hh e20 , ck 2 ; blind, > 64 eller paritetsfejl
      ca 254 , hh e20 ; blind hvis alle huller
a6: hv c5 , nt (e23) ; , caseshift
      pa e23 , hh e20 ;
a9: can s-e29+1, hv e6 ; indhop fra e4
      can s-d6 , hv e6 ; - indhop efter b
      ca se97 , hv a10 ;
a4h: hv e20 , ca 100 ; blind, skriv copy efter 1 el. fejl
a13: nc 100 , it -1 ; tæl ikke tomme linier
      qq (e2) , ga a13 ;
      sy (d3) , hv a12 ;

```

b a2 [input fra disc eller tromle]

```

e117:bt (-3) t 1 ; tæl karakter
      gm 6d3 , hv a ; næste ord
      arn -4 , ck -6 ; næste karakter
a2h: gr -4 , tk -4 ;
      ca 63 , arn a15 ; CR
ah: hv 1e20 , arn -5 ;
      pa -3 t -4 ;
      nc 39e50 , hv a1 ; hop hvis samme kanal
      arn 6d2 , ac -2 ; tæl kanaler
e101:is [kanal]t 1 ;
      bs s-449 t 510 ;
      pa e101 , it 1 ; næste gruppe
e100:vk [gruppe], vk(e101) ;
      lk e50 M ; læs næste kanal
      vk (e101) , it -39 ;
a1: pmm(-5) V LA t1 ; hent næste ord hvis tromle
      hs 20e50 ; ellers hop
      hv e121 NZ ; hop hvis baandfejl
      cl -6 , gm -4 ;
      pm 6d3 , hh a2 ;

```

e
e
e

[i = definition]

b a3

```

e53: hv c1 NZ ; index efter i
      ncn(e5) , hv c1 ; k definition udenfor blokhoved
      hs e11 , qq e4 ; læs def. adr.
      cl -30 , hv e42 ; sæt il

```

```

      [navn = def. adr.]
e12: it (e5) , pt a2 ; gem bogstav
      pa e52 , hs e10 ; redef. lovlig, læs def. adr.
a2h: pa e5 ;
      ps e4 , hv e51 ; hop til sporing

```

```

      [erklæring]
e85: it (4d3) , pa e5 ;
      bsn(e5) X t2 ;
      can p-1 , hv e26 ;
      hv c4 ;

```

```

e
=
      [sæt 1]
e42: gr a15 ; a15:= 1 ny
[+1] srm e47 , tk -20 ; 1bt
      ar a15 IPC ; iny-1bt
      tl -10 , tln -29 ;
      dln 1e62 , ar e47 ; R:= ent((iny-1bt)/40)+k0
      mb e90 , gr a14 ;

```

```

      [ændring]
e46: vke98[gruppe],vk(e44) ; vælg gruppe, vælg kl
      sk e39 , arn a14 ; skriv kl, R:= kny
      pa e45 ; frisk:= false
      hv e37 LPB ; indhop
e21: gr 1d5 , pm a15 ; kl:= kny, M:= iny
      gm d5 , ck -10 ; il:= iny, Radr:= kl
      ga e44 , srm e47 ; e44:= kl
      tk -20 , ar d5 ;
      tl -10 , tln -29 ;
      dln 1e62 X ;
      vk (e46) , vk (e44) ;
a10: pa e45 LZ t1 ;
      gr 2d5 , ck -10 ;
      ar 4d2 , ga e57 ;
      arn e44 , ca (b11) ;
      sk e38 , ud a3 ;
a16: lk e39 , vk 960 ;
e76: hr s1 , pp ; g vælg gammelt medium
      arn e111 , hh e41 ;

```

```

      [gemning]
e43: arn d ;
e57: gr [celle]MPC t1 ;
      arn 6d2 , ac d5 ; il:= il + 1
      ar 2d5 , pa e45 ; iv:= iv + 1, -, frisk
      gr 2d5 , sr 1e62 ;
      hr s1 LT ; udhop hvis kanal ikke fuld
      vk (e46) , vk (e44) ;
      sk e39 , arn 6d2 ;
e44: vk[kanal] t 1 ;
      ac 1d5 , hvm a10 ;

```

```

[understregning og betingelser]
e6: bs (2d3) , hv e94 ; hop hvis betingelse
    hs e20 , qq 6e4 ; læs
    ca 50 , psn e34 ; b begin
    ca 52 , hv 1e4 ; d blind
    ca 53 , psn e35 ; e end
    ca 54 , udn e62 ; f flydende
    ca 55 , hh e76 ; g gammelt medium
    ca 56 , hv e58 ; h hjælp
    ca 57 , hh e92 ; i informer operator
    ca 34 , psn e104 ; k karakter input
    ca 36 , pan e62 ; m maskintal
    ca 37 , pan e19 ; n normal ordre input
    ca 39 , psn e89 ; p parametre
    ca 41 , udn e56 ; r relativ ordre input
    ca 19 , psn e105 ; t text input
    ca 20 , hv e73 ; u udhops adresse
    ca 22 , psn e72 ; w sæt work
a2: ca s-6e4 , hv 1e4 ;
[1a2]arn s V LZ' ; kanal beskrivelse i R
    pm -2 V IPC ; eller gammelt medium i M
    ps s-1 , hv e ; hent kanal og hop
    qq V NPC ; skip hvis gammelt medium er ydre
    gm 40e50 MPC ; ellers gem gammelt medium
    pm e65 ;
    gm -2 M ;
    ca 35 , hh a4 ; l strimmel input
    sy 58 , sy 64 ; s skrivemaskine input
    pa -2 t 1 ;
a4:e75:pae2 , ud -2 ;
    pm e107 , gm e20 ;
e92h:hv 1e93 , it 1 ; , 1
    pa e2 , sy 64 ;
    sy 58 , hhn e91 ;

e56: pa e19 t 2 ; sæt r
e107:pp , lyn d3 ; e20 værdi ved læsning fra ydre enhed
e90: 1023 ;

b a2 [vælg discinput]
a2: gr (e31) , hv e31 ; gem et textord i stack
e89: qq c16 , it a2 ; modifier udhopsadresse
    pa e119 , it (e31) ; og gemmeadresse i textlæseprogram
a1: pa a , ud d6 ;
e97=-i, hs e20 , ps a1 ; læs char
    bs (e23) , hv r-1 ; skip chars in upper case
    nc 17 , hv e122 ; fortsæt hvis ikke <
e111:arn c14 D ;
e55:a: pp , hh e41 ; hent resten af vælg disc og hop
e

```

Slip karakterlayout.

		g 2d3								
		0	1	2	3	4	5	6	7	4d3
; y	4d3	0	1	2	3	4	5	6	7	
; 0	0	.	i	.--+	.10	.	. 5.0	.m	.	0
; 1	1	.	kB	.s	.	1
; 2	2	.	jA	.r	.	2
; 3	3	.	aK	.p	.	3
; 4	4	.	bZ	.t(.	4
; 5	5	.	cT	.	.	5
; 6	6	.	dM	.	.	6
; 7	7	.	eN	.	.	7
; 8	8D	.	.	8
; 9	9V	.	.	9
; 10	X	.	.	10
; 16	C	.	.	16
; 18	O	.	.	18
; 19	Q	.	.	19
; 21	L	.	.	21
; 26	R	.	.	26
; 34	P	.	.	34
; 44	I	.	.	44
; 45	nS	.	.	45
; 46	f)F	.	.	46
; 47	h	47
; 48	,/	.	.	.:	48
; 49	=	49
; 100	cr CR	.	.	.	100
; 101]	101
; 102	x	.	.>	102
; 109	ml ML	109
; 118		118
; 119	119.0	119
; 120	120.0	120
; 121	[.	121
; 122	;	122
; 123		123
; 124	sum	.	.clear	124
; 125	up LOW	125
; 126	126.0	126

; 119.0 blinde: stop end rød tab off on blaa
 ; STOP END RØD TAB OFF CN BLAA
 ; 120.0 forbudte: i lc: 10 15 26 42 45 46 47 65
 ; i uc: 10 15 26 42 45 46 47 65
 ; 126.0 blinde ogsaa i text: tf low TF UP
 ; 0.5 resten: aa uvwxyzloqøæg^ AA VUWYJØEEGH

[karaktertabel]

e1;;		lige		ulige				nr lige		ulige	
;		lower case	upper case	lower	upper			low	up	low	up
qq	0.32+109.39+	0.12+109.19+	0.2+	1.9+	5.22+	0.29;	0 ml	ML	1	V	
qq	0.32+ 2.39+	0.12+102.19+	0.2+	3.9+	4.22+	48.29;	2 2	x	3	/	
qq	0.32+ 4.39+	7.12+ 49.19+	0.2+	5.9+	0.22+122.29;	4 4	=	5	;		
qq	0.32+ 6.39+	0.12+121.19+	0.2+	7.9+	0.22+101.29;	6 6	[7]		
qq	0.32+ 8.39+	6.12+ 4.19+	0.2+	9.9+	5.22+ 46.29;	8 8	(9)		
qq	0.32+120.39+	0.12+120.19+	0.2+119.9+	0.22+119.29;	10				stop	STOP	
qq	0.32+199.39+	0.12+199.19+	5.2+	0.9+	5.22+ 0.29;	12 end	END	aa	AA		
qq	0.32+123.39+	0.12+118.19+	0.2+120.9+	0.22+120.29;	14						
qq	0.32+ 0.39+	5.12+ 0.19+	2.2+123.9+	1.22+102.29;	16 0		^	<	>		
qq	6.32+ 1.39+	5.12+ 45.19+	6.2+ 4.9+	5.22+ 5.29;	18 s		S	t	T		
qq	5.32+ 0.39+	5.12+ 0.19+	5.2+ 0.9+	5.22+ 9.29;	20 u		U	v	V		
qq	5.32+ 0.39+	5.12+ 0.19+	5.2+ 0.9+	5.22+ 10.29;	22 w		W	x	X		
qq	5.32+ 0.39+	5.12+ 0.19+	5.2+ 0.9+	5.22+ 4.29;	24 y		Y	z	Z		
qq	0.32+120.39+	0.12+120.19+	4.2+ 48.9+	3.22+ 0.29;	26						
qq	1.32+124.39+	1.12+124.19+	0.2+119.9+	0.22+119.29;	28 clear	CLEAR	rødd	RØD			
qq	0.32+119.39+	0.12+119.19+	0.2+119.9+	0.22+119.29;	30 tab	TAB	off	OFF			
qq	2.32+ 0.39+	2.12+ 0.19+	1.2+ 2.9+	5.22+ 0.29;	32 -	+	j	J			
qq	1.32+ 1.39+	5.12+ 3.19+	5.2+ 0.9+	5.22+ 21.29;	34 k	K	l	L			
qq	6.32+ 0.39+	5.12+ 6.19+	5.2+ 45.9+	5.22+ 7.29;	36 m	M	n	N			
qq	5.32+ 0.39+	5.12+ 18.19+	6.2+ 3.9+	5.22+ 34.29;	38 o	O	p	P			
qq	5.32+ 0.39+	5.12+ 19.19+	6.2+ 2.9+	5.22+ 26.29;	40 q	Q	r	R			
qq	0.32+120.39+	0.12+120.19+	5.2+ 0.9+	5.22+ 0.29;	42		ø	Ø			
qq	0.32+119.39+	0.12+119.19+	0.2+120.9+	0.22+120.29;	44 on	CN					
qq	0.32+120.39+	0.12+120.19+	0.2+120.9+	0.22+120.29;	46						
qq	5.32+ 0.39+	5.12+ 0.19+	1.2+ 3.9+	5.22+ 2.29;	48 æ	Æ	a	A			
qq	1.32+ 4.39+	5.12+ 1.19+	1.2+ 5.9+	5.22+ 16.29;	50 b	B	c	C			
qq	1.32+ 6.39+	5.12+ 8.19+	1.2+ 7.9+	5.22+ 0.29;	52 d	D	e	E			
qq	5.32+ 46.39+	5.12+ 46.19+	5.2+ 0.9+	5.22+ 0.29;	54 f	F	g	G			
qq	7.32+ 47.39+	5.12+ 0.19+	1.2+ 0.9+	5.22+ 44.29;	56 h	H	i	I			
qq	0.32+126.39+	0.12+125.19+	3.2+ 1.9+	7.22+ 48.29;	58 low	LOW	.	:			
qq	0.32+125.39+	0.12+126.19+	0.2+124.9+	0.22+124.29;	60 up	UP	sum	SUM			
qq	0.32+119.39+	0.12+119.19+	0.2+126.9+	0.22+126.29;	62 blaa	BLAA	tf	TF			
qq	4.32+100.39+	4.12+100.19+	0.2+120.9+	0.22+120.29;	64 cr	CR					

[begin]

```

e34: qq c11 , hv d6 ;
      arm a17 ;
      gr (e5) , it (a18) ;
b6: pt (e5) , hs e31 ;
      grn(e31) M ;
a18:e63: arm D t-1 ;
      hh b6 NZ ;
      hs e31 ;
      hv b1 ;

```

[sporing]

```

e51: ga e48 , arm d ; e48:= 11, R:= index
      hs e16 IOB ; hent, OB:= 0
      hv e52 LRA ; defineret
      pa a17 , pt a17 ;
b2: gr (e5) , it (e48) ; definer bit 1-10
      pa (e5) , hh 11 ;

```



```

b4:  arn a21      , ga  a20      ; næste til R
      ca          , hv  b3       ; færdig hvis 0
      ck          , tl  -30      ; id til pos 39
      ar a17      X              IOA ; k:= ent((Ref + id)/40)
      dln 1e62    , ck  -10      ; ks til Radr
      ca (e44)    , hvn b10      ; ks = kl
b11: ca[kanal]960, hv a19      ; ks = gammel ks
      vk (e46)    , vk (b11)    ; vælg gammel ks
[2b11]it(b11) t 512          ;
      bs 448      , sk  e38      ; skriv gammel ks hvis < 960
      ga b11      , vk (b11)    ;
      lk e38      , vk 960      ; læs ks
b10h:a19:arn a7, ar a16      ;
      ga a9       , cln -10     ;
      ac a9       , arn(a9)     ;
      ck -10      V            NOB ; if r v p then OA:= 1
      ck -2       V            NA  ;
      ck -2       IOA          ;
      ck 12       , ga  a21     ; næste adr gemmes
      gr (a9)     , it (e48)    ; c[id] ændres
      pa (a9)     , ck 29       ;
      pi         LO t511       ; if p then OA:= 0
      tk 31       ;
e49: bs          , hhn b8      ; ikke h
a24: qq          V            NOB ; p, tælledele
      hhn b8      NA          ;
      ck -4       , pm 6d2     ;
      ca 56       , tl 28      ; HV -> HH
      ca 32       , tl 26      ; PA -> PT
b8:  tl 27       , ar (a9)     ;
a20: sr          D            LOA ; -r
      ck -10      LOB          ; 10 - 19
a9:  gr          , hv  b4      ; paa plads
b3:  arn(e5)     , ck 10       ; ordbog
      pi 256      V NOB -257   ; 0 - 9 eller 10 - 19
a7:  qq e38-e39v ;
a21: qq          , hv  b2      ;
a1:  ck 20       NRB          ; ordbog paa plads
      gr (e5)     V            MQC ;
e74: nc (e48)    , hv  c6      ;
      hr s+1      ;

```

[redefinition]

```

e52: bs          , hv  e74      ; kontroldefinition
      gr (e5)     , it (e48)    ;
      pa (e5)     , it (e48)    ;
      pt (e5)     , arn(e5)     ;
      ck 20       , hv  a1      ;

```

[tæl op i stak]

```

e31: it e71      t 1          ;
e116:bs e39-1    , hr  s1      ;
e58h:hv c9       , pa  e110    ; stakoverløb, h udhop til help
      pi 16       , hv  e46     ;

```

```

[udhopsadresse og end]
e73: ps e4 , qq d6 ; u, ved e hoppes retur til e35, d6 til s
e35: e109: qq c12, pm (2d) ; e endkanalnr. R40, 41:= ref40, 41
      hsn e20 X IPC ; Indicer ref i P, læs char
      ca 64 , hv s ; hop hvis CR
      hs e9 ; læs def adr
      ck 10 ; enhed i pos 39
      gr e70 MB ; sæt udhopsadr i exitparam
      hv s IPB ; indicer udhop i PB og hop

e61: gr a14 , tln 10 ; resten af end
      gr a15 , hv e46 ;

e3: [bufferord eller kanalvalg]
<d35, qq e38+40.19+201.39 ; 40 ord til e38
      qq d6+40.19+201.39 ; 40 ord til d6
x ga 3e3 ;
[1e3]vk V NT ; vælg gruppe
[2e3]vk -1 , it -64 ;
[3e3]vk , hr s1 ; vælg kanal
>

[hent kanal til d6]
e36: arn c18 D ; CP-kanal
e: ca c18 , hr s1 ; udhop hvis samme kanal
<d35, ga e , tk -31 ;
      ar 1e3 , il ; læs fra buffer
      hr s1 ;
x hs e3 ; vælg kanal
      ga e , lk d6 ;
      vk (3e3) , hr s1 ;
>

[beting]
e94: hs e10 , qq 6e4 ;
      bs (1d1) , hv e4 ;
      pp 302 , hh e20 ;

e72: qq c10 , hv e86 ; sæt work
e65: vy e103 t -8 ; e55 værdi ved læsning med 1
e104: qq c16 , hv d6 ; karakterinput
e105: qq c16 , hh e54 ; textinput

e37: vk (b11) , ud 2b11 ; afslut slip
      sk e38 ; skriv sporingskanal hvis nødvendigt
      vk 960 , vk d11 ;
      sk e110 , arn -2 ;
      qq (-5) LA td14-e50 ;
      hv -9 ; skriv paramkanal, og hop til help.

e110: qq 2 ; udhopsparametre til help
      textit; ;
e70: qqf , ; her sættes evt. udhopsadresse
      qqf , ;

```

```

ba      [tallæsning og fejlreaktioner]
e28: arn e60 , hs e ;
c1h: hv d6 , hs a ; 1. syntax læs til ,/: e1 CR og hop e4
c2: hs a , hv e4 ; 2. forb. udef. samme reaktion
c3: hs a ; 3. ikke erklæret -
c4: hs a ; 4. gal erkl. læs ,/: e1 CR og hop 1b1
c5: hs a ; 5. ubenyttet symb. fortsætter
c6: hs a , qq e4-1 ; 6. kontrol def. -
c7: hs a ; 7. talfejl syntax -
c8: hs a ; 8. - størrelse -
c9: hs a , qq a2 ; 9. stak overløb læser fra skrivem.
e121:hs a , qq a2 ;10. medium eksisterer ikke -

```

```

a: arn c13 D ; fejl
[hent kanal til e38]
e41: ar e108 , vk (e46) ;
vk (b11) , ud 2b11 ; valg evt. sporingskanal, test om i brug
e60: qq c15 , sk e38 ;
a3:[1e60] pa b11 t 960 ;
e120:<d35,tk -31 , ar e3 ; dan bufferord
vk 960 , il ; læs fra buffer
x hs e3 ; valg kanal
lk e38 , vk (3e3) ; læs kanal og vent
>e108:qq c10 , hv e38 ;
e

```

```

b a21 [tallæsning]
a6=e95+e123-2

```

```

a17: hsn e20 X ;
a3: pa a7 IZB ; sæt ikke decimaler, og førcifre
ca 32 , it (e23) ;
pa a4 , ca 32 ; sæt fortegn
a8: hs e20 IZB ;
bs p , hr s+1 ; udhop hvis ikke cifre
ARS 4d3 , CK 10 ;
a7: PP , ML a6 ; tal:= tal × 10 + ciffer
PP p1 NZ ; korriger for flere cifre
d1 a6 X NZ ; end der kan være i M
CA Sp , HV a8 ; ingen faktor korrektion
GM d , PM d1 ;
a15: MKS pa5 , NK a14 ; .8 eller 10/16
BS p , IT 4 ;
a14: IT t-3 ;
a2: XR , IT -1 ; korriger exp2
a1: BT , HV a15 ; kun effektiv under mpotens omr.
GM d1 , PM d ;
CAS s-e , HV a16 ; udhop efter potens omregn.
a9: HV a8 , GM d ; her starter læsning af mpotens
HS a17 ;
PP (a4) X 9 ; if pos exp then p:= 1
BS p , PP 1 ; else p:= -1
PM d1 , SR 2d2 ;
HS c8 NT ; fejl exp for stor
AR 2d2 , CK -10 ;
GA a1 , HH a2 ; sæt exp 10; hop til potens omr.
a19: PA a18 t 40 LZB ; Gruppe læsning
ARS d , TK 10 ;
GR d , HS a17 ;
XR , MT a4 ;

```

[30.5.67

slip, page 15]

```

a18: TK t-10 ;
      NS (a18) , CK s ;
      AC d X ;
      CA 3 , HV a19 ; ny/
a13: PP (2d3) , ARS a5 ; Talslut,
      BS p-1 , HV a ;
      IT (4d3) , CK (4d3) ;
      GA a20 , IT -49 ;
a20: PI , PP 4 ; sæt mrk.
a: HS e43 ; sliplagring
      BS p+508 , HS c7 ; fejl terminator
      PM d3 X ;
a21: CA 64 , HV e4 ; CR, udhop
      HS e20 , QQ e ; læs evt. blinde terminatorer
      HV 1d6 ;

<d35, b a9, b4 [initialicer slip]
a6: qq 1i+2.19+1546.39 ;
a7:c38:arn c37-5, ar c37-4 ; dan bufferord
e71:a8:tk 20 , ck -20 ;
a: ac e3 , ac 1e3 ;
      ac a1 , grn a ;
a5: arn c37-4 , ar 6d2 ;
      hh a9 NZ ;
      arn a6 , il ;
      arn a8 , ac a7 ;
      ac c37-5 , arn a6 ;
a9h: us , hs c71 ; get word
      xr V LZ ;
      hv c70 ; baandfejl
      ar 2 D LA ;
      ar 1 D LB ;
      ac a , it -1 ;
      bt e67 , hv a5 ;
<d35-2,us(-31) t -96> ; rewind
<d35,arn a ;
      hv c77 NZ ; sumfejl
      arn a1 , us ; gem op tabel i buffer
x b a4, b4 ;
a: vk V ;
c38: arn c37 , ar a1 ; beregn abs kanal af sidste kanal i slip
e71: tl -16 , tln -23 ;
      dl b , ar b ; e71 = staktop ved indhop,
      ck -10 , ga 1e3 ; sæt gruppenr
      ga a X ;
      sr a1 , pm a ;
      gm 1e3 V NT ;
      it (a) , qq (2e3) ;
      ck -10 , ac e72 ;
      ac e111 , ac e108 ;
      ac e34 , ac e109 ;
      ac e36 , ac e89 ;
      ac e104 , ac e105 ;
      ac e60 , ac e ;

```

>

[27.11.67

slip, page 16]

```
a3:  grn e71-2 X      MC ; sæt slut mærke i første celle af stak
      gm d3-1 M      t1 ; nulstil arbejdsceller
      bt 26      t    -1 ;
      vk 960      , hv a3 ;
      vk d11      , lk e38 ; læs parameterkanal
      vk d37+d9,  , arn 1e38 ; hent parameter
      ar 960      D      NT ; adder 960 til gruppe nr hvis mindre end 512
      arn a2      V      LA ; hvis parameter ≠ helpnumber hent image
      arn a2      NB      ; beskrivelse
      gm e71-1 , ga e46 ; sæt gruppe nr i e46
      ga b11      , it (b11) ;
      pt 1e60      ;
      tk 10      , gt e47 ; ibt.19 til e47
      gt 2d      , cl 10 ; og 2d
      gm e66      , gt d ; kb.39 til e66, i.19 til d
      sr d      , ck 10 ; ibt.39 ren til R
      gr 1e66      , mt b4 ; ibt.39 til 1e66
      ml 1e62      , gm e30 ; 40 kb-ibt til e30
      arn e66      , pm b2 ;
      gm 40e50 M      ; sæt ingen gammel streng
      ac e47      , lk e84 ; ibt.19+(40kb-ibt).39 til e47, læs text kan.
      vk 2d9      , lke39+e123 ; læs taltryk kanal
      hs e21      , ps 512 ; start tromleblok, 512 til s
      gs 3d5      , gs 4d5 ; 512 til 3d5, 4d5, ..., 7d5
      gs 5d5      , gs 6d5 ;
      gs 7d5      , arn d ;
      pa 2d      IPC te 71-2 ; 2d:= e71 - 2.9 + ibt.39
      pa e47      IQC te 71-1 ; e47:= e71 - 1.9 + ibt.19 + kb.39
b4:  ck -20      , hs e42 ; sæt il
      hv b1      ;

b2:  pa e92      V te 121 ;
a2:  qq e98+294d10.19+10.39;

a1:  b3=401-j, c=1k

<d35, d4=-480, e40=80, c39=1c-c30
xd4=6c-c30, e40=1
>
```

b k=c, i=e38, a7 [mediumvalg]
 $\bar{c}14=d4$, $d4=d4+e40$, $e69=k$

```

arn -2          IPC ;
qq             V   NPC ;
gr 40e50        MPC ; gem gammelt medium hvis internt
bs p1           , hv a       ; = gammelt medium
it (e111)       , pa e       ; indicer d6 tom
e87: ps r-1      , arn [e59] ; fyld sidste ord i navn op
hv [e122]       NOA ;
hs [e122]       ;
vk 960          , it e50     ;
bs (e31)        , hv c9      ; ikke plads i lager
gp e31          , pp p-1     ; afstak navn
vk d37+d9-1, lk d6 ; læs searchkanal
pmm c36 DX      IZA ;
vk 960          , it e50     ;
pa c35+d6, hs c32+d6; search for mediumname
arn 1.2 VD      LZ ;
hv e121         ; name is not in katalog
ar 2d6          , vk 960     ;
hv e121         LT ; fejl hvis outputmedium
arn 2d6          , vk c63     ;
qq e50          , lk d6      ; læs init medium
vk 960          , hs d6      ; init medium
hv e121         NZ ; baandfejl
a6: pm -2       IQB ;
hv a2           NC ; hop hvis ydre medium
pa e116 t 1e 50 ;
dln ra4         , ar ra4     ;
ck -10          , ga e100    ; gem gruppenr
vk (e100)        , cl -10    ; vælg gruppe
qq (-5) NQC te50-d14; kanalplads til e50
a3: ga e101      , vk (e101) ; gem og vælg kanal
a7: lk e50       , vy 16     ;
pm ra5          , vk 960     ;
gm e20          , hv 1e93    ;
a: pm 40e50      IQC ;
gm -2           MQC ;
hh 1a6          ;
a2=e75
a4: 960         ;
a5:c=1k, pp     , hv e117 ;

```

<i-40e38, >; længde af vælg medium
 $b=a4-e38+b3$, $b1=a6-e38+b3$

e
 e
 e

[her kan indskydes flere kanaler]

<d35, x d4=0, c39=c-c30, c=c30-6>

b k=c, i=d6, b7 [tallæsning]
c15=d4, d4=d4+e40

```

ARS d3      , HH  b      ;
BS  p+511 , HV  b1      ; test for blinde,
BS  p+510 t  -511      ; terminatorer
b:  hv  a21  , PA  a1      ; sæt exp10
b1: PM 256   DX      IZB ; sæt cifre ikke læst
GR  d1      , PA  a2      ; sæt 10-potens faktor og exp2
HSS a3      X          ; læs evt. fortegn og cifre
BS  p509 -512      IPC ; sæt NPC, test for . og 10
XR   , HH  b2      ; rent heltal læst
BS (e23) , HV  b3      ; 10 uden.
pa  a7      t  -1      ; sæt efter .
HS  a8      ; læs decimaler
HS  c7      NZB ; fejl . uden cifre
b3: PM 6d2   NZB ; 2^(-39), 10 uden cifre
a4: XR      , MT  a4      ;
PM  e23     X          ;
CA  p+7     , HH  a9      ; 10
a16: MLS d1  , NL  b7      ; heltal x 10 potens faktor
HV  b4      LZ          ; tal = 0;
b7: IT      , PP (a2)      ;
BS (e62) , HV  b5      ; flydende
TL  1      ITA ;
TK  -1     , GR  d        ;
AN  nd     NTA ;
BS  p+39   , HS  c8      ; fejl, maskintal for stort,
TK  p+40   ;
SR  6d2    LO ; korriger +1,
b4: GR d    V          ; maskintal lagres
b5: NKF p+40 , GRF d      ; flydende tal lagres
HS  c8      LO ; fejl, maskintal for stort
b6: hs c7    NZB ; endelig lagring
b2: hv a13   , MT  a4      ; fejl, uden cifre
GR  d      X          ; heltal
CA  3      , HV  a19      ; test for gruppe
BS (e62) , HV  a16      ; test for f, i = 370
BS (a2)  , HS  c8      ; fejl, heltal for stort
HV  b6      ; hop til test for cifre og lagring
can s409 , cm (r-410);
a5: qq 1.10+7.15 ;
c=1k, qq 320 ;
<i-40d6, :>; længde af læs tal

```

e
e
e

```

[begin]
b k=c, i=d6, a5
c11=d4, d4=d4+e40

hs e20 ; læs tegn, understregning inform
ca 64 , hv e4 ; , ny linie (ordre)
b12: ca 34 , pi 32 ; , k
arn 1d5 ;
gr a14 , tk 10 ; Kny:= Kl
ar 2d , ar d5 ;
gr (e31) MPC ; pak[ref, ib, kl, il]
arn d3 , ca 34 ; , k
ps a3 , hv a4 ;
a1:[9d6] ca 57 , hh a ;
b17: arn d5 , hh a5 ; Iny:= Il
b9:a4: can(e45), arn 6d2 ; if -, frisk then Kl:= Kl + 1
b18:a3:ac 1d5 , hv e13 ; læs = def. adr.
cl -30 , gr a14 ; kny:= læst
hs e14 , hv b17 ; if CR
a: hv a1 , hs e13 ;
b19h:a5h:cl -30, gr a15 ; il:= Iny
tk 20 , gr 2d ;
it (e31) , pa 2d ; ref:= n
hs e31 , qq a2 ; tæl n
hv 1e42 NPA ; ikke tromle
arn e47 , gr (e31) ; gem tromleref
arn a14 , ar 2d ;
gr e47 , it (e31) ;
pa e47 , hs e31 ; pak[reft, ibt, kb]
a2: hs e46 IPC ; ændring
arn e47 , tl -30 ;
tln 10 , tk 19 ;
tl -19 , mt r ;
ml 1e62 , gm a17 ; Ref:= kb x 40 - ibt
b1: can p-4 ; test
hs e14 , hv 1e93 ;
hs e85 ; max index i R
pm (e5) X td5 ; R:= stak[i], e5a:= i
gr (e31) , sr 2d ; stak[n]:= stak[i]
hv c4 NT ; if start[i] > ref d.v.s. dobbelt erklæring
ncn p-4 , hv c4 ;
it (e31) , pa a17 ;
cln -11 , ar e15 ;
c=1k,ga e63 , hv 1e34 ;
<i-40d6,:>; længde af begin
e

```


[30.5.67

slip, page 20]

b k=c, i=d6, b7 [end]
c12=d4, d4=d4+e40

gk e4 , ps b2 ;
vy , arn e47 ;
b1: vy 32 LKA t-33 ;
vy 16 LKB t-17 ;
sy 64 , ud e79 ;
b2: hh s , hs e95 ;
arn 2d , hs e80 ;
arn d5 , hs e81 ;
arn 1d5 , hs e81 ;
[330]pt e18 t b5 ;
pp 2 , it (2d) ;
b6: pa e31 , arn p1d5 ;
ga a24 , pp p1 ;
gan b , it (a24) ;
bs (e31) , hv b4 ;
b3: gp e5 , hs e16 ;
gr d , vy ;
hv r4 LRA ;
tl -20 ;
[340]vy 16 NZ ;
hv b LZ ;
hs b1 , sy p46 ;
arn b , hs e33 ;
arn d , hs e77 ;
hv b LRA ;
arn d , hs e80 ;
b: arn D t1 ;
ck -30 , hv b3 ;
b5: arn(a24) , gr pd5 ;
b4:[350]bs p505, hh b6 ;
pt e18 t e5 ;
hs b1 , arn(e47) ;
gr e47 LPA ;
arn d5 , gr a15 ;
arn 1d5 , gr a14 ;
arn(e31) , tl -20 ;
tk 20 , gr 2d ;
ps e93 , vy (e4) ;
hv e46 NPA ;
c=1k, tln 10 , hv e61 ;
<i-40d6,:>; længe af end
e

b k=c, i=d6, a16 [text]
 c16=d4, d4=d4+e40

```

a16: pm 32      D          ; k karakterinput
      pt a6      t      a  ;
a:   hs e20     , arn d3   ; læs eller hent symbol
      ca 19     , hv a1    ; hop hvis t eller T
      ca 59     , hv a15   ; hop hvis punkt
      bs (2d3)  , hv a     ; hop hvis ≠ 0-9
      gm d      , it 9     ;
      bs (4d3)  , hv a     ;
      hsn e24   X         ; læs karakter nr
      tk 30     ;
      pm d      , hs a5    ;
e54: hh a       , ud a16   ; , t textinput
      pt a6     Vt a11    ;
a1:  bs (e23)   , it 1     ;
a9:  pa a8      , pp e82   ; output-case:= lower, text værdi i p
      ps 1      , hh e20   ; læs et symbol
      bs (e23)  , hv a6    ; hvis input-case = upper, hop
a8:  bs[output case],hh 11 ; hvis output-case = upper, hop
      hh a7     , pa a8    ; , output-case:= lower
      arn 58    D         ;
a11: hh a10     , ud 1-1   ; , afslutning
      bs (a8)   , hs a5    ; hvis output-case = upper, output lower
a15: ps r-1     , arn a13  ; fyld sidste ord op
      hv a5     NOA       ;
      hs a5     , qq e4-1 ;
      hhn e91   ;

a6:  ca 5       , hh a11   ; hvis ; , hop
      bs (a8)   , hh a7    ; hvis out-case = upper, hop
      arn 60    D         ;
a10: ga a8      , hs a5    ;
a7:  arn d3     , ps a9    ;
a5:e122:ar a12  , ca 64    ;
      ar -1     D         ;
      cl 36     V         LOA ;
      ck 10     , cl -6    ;
e119:hs e57     LOA       ;
      cln 7     LOA       ;
      hr s1     IOA       ;

a12: 64//15     ;
a13:e59:
c=1k, qq 59.39+10 ;
<1-40d6,:>; længde af text
e
b k=e69, i=e38
I=e87, ps r-1 , arn e59 ;
      hv e122   NOA      ;
      hs e122   ;
e

```

[27.11.67

slip, page 22]

b k=c, i=e38, a8 [fej1]
c17=d4, d4=d4+e40

```
      can s-1      , hv a5      ; sumfej1
a1:a2:gi a        , gk r        ;
      pi          , sy 58        ;
      pi (a1)      IK t-8        ;
      ncn p-1      , it 3        ;
      pa e2        , sy 29        ;
      arm s-c2+2D  IK          ;
      ga a2        , gs a3        ;
      sy          , hs e79        ; tryk fejltype
      arm d5        , hs e81        ; tryk adresse
a:    pi          , sy          ;
      can(e2)      , hv a4        ; hop skrivemaskine-input
[610]bs (e23)     , sy 60        ;
      sy (d3)      , sy 62        ; tryk sidst læste symbol
a3:   ps          , pm 1d        ;
      pa e92       t 1e 93        ;
      bs s-c8      , arm 1a1      ;
      bs s-c5      , hr e15        ; udhop for type 6, 7, 8 og 9
      bs s-c4      , hr 6e4        ; udhop for type 5
      bs s-c3      , it b1        ; sæt udhop for type 4
      ps          t e4 -1        ; sæt udhop for type 1, 2 og 3
      pp 248       , hh e20        ; skip til ,/: el VR
a4:   pa e23       t -10         ; sæt lower case hvis
      hh a3-1      ; skrivemaskineinput
```

```
      [sumfej1]
a5:   hs e84              ; output text sum
      qq a6          , arm 2d3      ; hent den korrekte sum til R
      ck 3           , sy          ; tryk bitmønster
      pa a7          t 6          ;
a8:   sy 1           V          LO ;
      sy 16          ;
a7:   bt 6           t -1         ;
      ck 1           , hv a8        ;
      pa e2          t 1          ;
      sy 62          , pm 1d        ;
      ps (4d3)       , hh e20        ;
```

a6: c=1k, k 64, 29, 58 tsum;.
<i-40e38,:>; længde af Fejl
e
-

b k=c, i=e38, a8 [sum-check]
 c10=d4, d4=d4+e40

```

      bs (2d3) , hv a ; hop hvis clear
      gm 1d , arn e96 ; gem M, hent sum
      tk -5 , ar e96 ;
      mb a8 , ar a8 ;
      ga 2d3 , pm 1e20 ; sum rest til 2d3
      arn a1 , gr 1e20 ; ret ord i 1e20
a: pa e96 , hh e20 ; sum:= 0, læs
a1: hh a1 , gm 1e20 ; tilbage fra 1e20, retabler 1e20
      ca 64 , ar a2 ;
      nc (2d3) , hv a4 ; hop hvis sumfejl
      pm 1d , hh e20 ;
a4: gs 4d3 , ps 1 ;
      hv 1e121 ; hent sumfejlkanal og hop

```

```

a8: qq 31 ;
a2: qq 1 ;
a7=i-e38 ;

```

b k=c, i=d6, a1 [sæt work]
 i=a7+d6

```

e86: vk (e46) , vk (b11) ;
      it (b11) t 512 ;
      bs 448 , sk e38 ; skriv sporingskanal
      vk 960 , vk d21 ;
      lk e38 , hs e10 ; læs katalogkanal, læs antal
      ga a , hs e10 ; læs beg kanalnr
      ck -16 , ar a ; pak work ord
      ck -14 , vk 960 ;
      vk d21 , sr 3e38 ; beregn ændring af work
      ac 3e38 , sc 39e38 ; sæt work og sum
      pa b11 t 960 ; indicer e38 tom
      sk e38 , hv e4 ; skriv katalogkanal
a1=e98-960 ;
a: c=1k, qq 4d32.31+a1.15-a1.19

```

<i-39d6,:>; længde af sumcheck og sæt work
 [sumcelle]

e
e

c13=c17-c10, c18=d4

<d35, c31=c-c30, c=c31-c39, c=c+c+c+c+c, c=c+c+c+c, e67=c+c-1
 qq d6+40.19+402c18.40

x c31=6c39

qq c31.39-1.39

i=c30, c;; galt antal kanaler

>

e [slip]

c30=c30-d1

b k=d42, i=0, a6 [load slip]
 T=d2

```
<d35      [buffermode]
vk d50    , vk d1+c39; dan checksum af løse kanaler
a: lkn d49 , pp      ;
vk d1+c39t 1      ;
it (a1)    , pa a2   ;
a1: nt d49  , lkd49+40d49;
a2: ar p    , pp p1   ;
ar 2      D      LA   ;
ar 1      D      LB   ;
bs p472   , hv a2    ;
bt c31-1c39 t -1    ;
hh a      ;
pp p-1    , sc (a2)  ;
vk d1+c31-1, pp      ;
sk (a2)   , vk 960   ;
```

<d39, <d35-1, <-d35+3; beregn antal karuselblokke
 pmm c31.7+c31+63 D

d1 64 D

x <d39, <d35; beregn antal disc eller baand blokke

pmm 9c31 D

d1 10 D

> <d39, <d35, gr a3>

<d39, hs 1

hv a4

<d36,

tres;

qqf c31.39

x<d39,

tset;

; set slip in cat

qqf d35.39

a3: qqf c31.39

[STOP, SUM]a i base of slip

s [STOP, CLEAR] > <d39

qq 39,

qq 57,

qq 52,

qqf

tslip;

qq c39+c38.19+d20.29+c30.39

qqf,

a4: hs 1

hv a5

tmove;

d=d19-960

qq 50,

qqf c31.23+d1.39+d.29-d.33

tslip;

qqf,

a5: hs 1

hv a6

tsetsum;

tslip;

qqf,

d2=1, a6: hsf 2

[30.5.67

slip, page 25]

x d2=1, hv d47

i=d48

d=d50-960, qq d35.2+d36.5+11.7+c31.23+d1.39+d.29-d.33,

qq,

tslip;

qqf c39+c38.19+d20.29+c30.39

d48=1, qqf

> d1=d1+c31

e

[STOP, SUM]a i slip

s

[STOP, CLEAR]
[23.6.67

(2)

start, page 1]

```
b k=d1, i=40d13, a5, b5 ; begin start

b=80d13 ;
<d35-2, us(-31) t - 96 > ; ENTRY START: if aux kind = tape then rewind tape;
vk d19 , vk d16 ; read image track 0;
lk b , vk d16 ; core init:= full init;
pa 8b , sk b ; write image track 0;
vk 25d16 , lk b ; read image track 25;
vk 25d16 , grn 23b ; clear core inhibit;
sk b ; write image track 25;
ps 40 , pm b3 ; for i:= 40 step -1 until 1 do
a: gm s39b , MB ;
ps s-1 , bs s ; buf[1]:= hsf 2;
hv a ;

a2h: pp d13-1 , hs c52 ; NEXT PARAM: get param;
hv a1 , hv c58 ; if number then goto set date; if single then alarm;
hv -9 , tl -7 ; if end list then return to help;
nc 0 , hv c58 ; if kind = drum then param alarm;
tl 5 , ca 1.5+d32.7 ; if work then
arn d14+d45, tl -2 ; R:= work-as-output;
tl -30 , tln 16 ;
gr b2 , tln -23 ; blocks:= bits 8:23;
hs c3 ; select track (bits 24:39);
a3: sk 40b , arn b2 ; rep: write to drum(buf)
sr b1 , gr b2 ; blocks := blocks - 1; marks:= 11;
hh a2 , LZ ; if blocks = 0 then goto next param;
ps a3-1 , hv c16 ; select next track; goto rep;

a1: arn 2c18 D IZA ; SET DATE: ZA:= 0;
hs c2 ; get first track of catalog;
grn d14+d51, arn p1 ; run:= bits 0:9 of param;
ga d14+d51, sr (r) ;
gr d14+d51, it 4.2 ; date:= qq 4.2 + bits 10:39 of param;
pa d14+d51, hs c8 ; correct sum;
sk d14 , hh a2 ; write track back on drum; goto next param;

b1: m 1c ; count
b2: qq [no of tracks] ;
b3: hsf 2 ;
```

```

i=i+39, d=k-d1          ; d = no of tracks
b k=d42, i=0, a10       ;
a1=d19-960              ; a1 = group no for image
a=d, <d35, a=1>          ; a = no of blocks

<d39                     ; if aux only then
i=d2, hs1                ; begin
    hv a4                ; (if aux reserved then
<d36,                    ;
    tres;                ;
    qqf a.39              ; res, no of blocks,
x <d39                    ; else
    tset;                ; set, aux kind, no of blocks,
    qqf d35.39            ; type in)
    qqf a.39              ;
[after i follows STOP, SUM and a sum character]
ia base, start
s
[STOP, CLEAR]
>< d39                    ;
    qq 39,                ; concat pid 0,
    qq 57,                ;
    qq 52,                ;
    qqf                    ;
    tstart;               ; start, spec <
    qqf,                  ;

a4:  hs 1                  ;
    hv a5                  ;
    tmove;                ; move, blockplace, start <
    qq 50,                 ;
    qqf d.23+d1.39+a1.29-a1.33 ;
    tstart;               ;
    qqf,                  ;

a5:  hs 1                  ;
    hv a6                  ;
    tsetsum;              ;
    tstart;               ;
    qqf,                  ;

d2=i                      ; end else
a6:  hsf 2                 ;
x                                         ;
i=d48                               ;
    qq d35.2+11.7+d36.5+d.23+d1.39, ; load to primitive catalog;
    qq,                     ;
    tstart;                ;
d48=i, qqf                  ;
>                               ;
d1=d+d1                      ;
e                               ; end image load
e                               ; end start
[after i follows STOP, SUM and a sumcharacter]
ia start
s

```



```
;slip<
[3.7.67]
[STOP,CLEAR]
```

[system punch/special binin]

[page 1]

```
b i=10,b50,c20,d10,e10
  hv rc15
b a20
d e0=552,e1=41e0,e2=41e1
```

```
; program start
; special binin block
; lower buf, upper buf, save catalog
```

```
c0: pmra1      , gm s1
a:  ar r0      Vt 1 IQC
a1: gm s3      t -1 M
    ar 2       D   LA
    bs(ra)     , hv ra
a2: tk 1       , ps (10)
    vy 17      , gk rb19
    pa 17      t 4
    hv 17      NZ
d0: vy[d18+d17], girb19
d4: vk[d21]    , lk e2
    ly rb      , ly rb
[-1] ly rb     , ly rb
    nc 13[aa]  , hh r-1
a3:b: lyn[char] D
    ac (rb1)   DV   NT
    hv ra10
    qq (rb2)   t 1
    ca 66      , hv ra11
    pp 39      , ca 40
    pp -1      V
    nc 1       , hv ra10
```

```
; reestablish primitive input;
;
;
; R:= checksum;
; prep sum error: restore s;
; select(t,w), save tk;
; prepare SUM message;
; if R=0 then goto write SUM;
; save indicator;
; save date and run no;
; SKIP CHECKSUM:
; for i:= lyn,lyn,lyn,lyn while i<aa do;
; INCHAR:
; Raddr:=lyn;
; if Raddr<0 then goto error else
;   sum1:= sum1+Raddr;
; sum2:= sum2+1;
; if Raddr=66 then goto FINISH TAPE;
; if Raddr=40 then p:=39 else
;   if Raddr=1 then p:=39 else
;     goto error;
```

```
a4: pmn 1.3    DX   IZA
a5: tl -7      , ly r1
[1] pi [marks] t 12 LZA
b1: ac [sum1]  DXV  NT
    hv ra10
    hv ra5     X    LZ
b2: qq [sum2]  Xt 6
    tl 3       , pp p1
b3: gr pe0     MRC
    bs p473    , hv ra4
    bs p-39    , hv ra9
    pm rb6     V    NQA
    arn rb10   , hv ra7
    dlrb7      , ar rb7
    ck -10     , ga rb4
b4: vk [group] , cln -20
b5: gt rb5     , vk [track]
    is (rb3)   , sk s-39
a6: arn rb8    , ac rb6
    nt -41     , it 0
    qq (rb3)   , hv ra3
```

```
; INWORD: RM:=0; LZA:=true;
; for i:=1 step 1 until 6 do begin
;   RM:=RM:128/lyn; if LZA then begin
;     RC:=bits 8 and 9; LZA:=false end;
;   if Raddr<0 then goto error else
;     sum1:=sum1+Raddr end;
;   sum2:=sum2+6; p:=p+1;
;   cell[buf+p]MRC:=RMx8;
;   if p<39 then goto INWORD;
;   if p=40 then goto AREAWORD;
; NEXT TRACK:
; if disc then goto outtrack;
; vk(track:960+960);
; vk(track mod 960);
; write track from(buf);
; track:=track+1;
; buf:= if buf=lower then upper else lower;
; goto INCHAR;
```

```
b6: qq [current track]
b7: qq 960.39
b8: qq 1.39
```

```

d1:b9: qq [d53.9]+0.39      ; parameterwords for
[1b9]  qq [d53.9+53.39]    ; us [unit]

b10: qq e0.9+40.19-40.39    ; parameterwords for
[1b10] qq e0.9+40.19-40.39 ; us 0

b11: qq 40.39                ;
[1b11] qq 1.21                ;
                                ;outtrack:
a7:  ar rb11      , us        ; param:=param+40; us(0,buf,param);
d2:b12:           ; count:=count+1; ....
      bt [count] t -1        ; if count>0 then goto INCHAR;
      gr rb10      , hv ra3    ;
d3:b14:           ;outblock:
      pp [d53:40-1], grrb10    ; count:=block:40-1;
b13: can +1.1      , hs ra8    ; if -,first then sense;
b17: nt 1b9-i      , amrb9+1b9; exchange buffers;
      ar rb6        , us (rb18); us(unit,track,buffer);
      nt 1b10-i     , pmrb10+1b10;
      gm rb10       , pa rb13   ; param:=new buffer base-40;
      amrb1b11      , ac rb6    ; track:=track+1;
      gp rb12       , hv ra3    ; goto INCHAR;

b15: qq                ; status word
[1b15] qq b15.9+1.19+4095.39; get status
                                .....
                                ; procedure sense;
a8:  pa rb16      , amrb1b15    ; begin rep:=0; ....
b18: il [unit]    , il          ; L: if status word(unit)<0 then
      amrb15      , pm rb6      ; begin
      hv s1        NT          ; rep:=rep+1;..
b16: bt [rep]     Xt -150       ; if rep>3 then
      hv ra10      ;           ; goto error;
      pa r1        , it (rb17)  ; us(unit,track,buffer);
[1]  is r0        , ar si-b17   ; goto L
      us (rb18)    , hh ra8     ; end
                                ; end;
                                .....
a9:  hs ra8        LQA          ;AREAWORD: if disc then sense;
      pm (rb3)     X   IQC      ; R:=set marks(cell[buf+p]); disc:=LA;
      pa rb3       Vte0 LQA     ; cat:=LB; if +;disc then begin
      gr rb6       , hv ra3     ; tracks:=R; goto INCHAR end; ...
      ga rb18      , tk 18      ; unit:= part1(R)+16; track:=Rpos21;
      it (rb18)    t 16         ;
      pa rb13      , gr rb6     ; first:=true; ....
      vk 960       , hv ra3     ; wait for track; goto INCHAR;

```

```

a10:
b19: pi [in]      , vk [tk]      ;error: restore indicator; restore tk;
      amnr        , hv ra2      ; R:=not zero; goto prep sum error;

a11: lynrb       , tk -7        ;FINISH TAPE:..
      ly rb       , tk -7        ; R:=((lynshift-7/vlyn)shift-7
      ly rb       , tk 4         ; ..vlyn)shift 4;
      nc (rb2)    , hv ra10      ; if part1(R)≠sum2v...
      ck 10       , vk 960      ; part2(R)≠sum1 then
      nc (rb1)    , hv ra10      ; goto error; select group(0);

      vk (rd4)    , ps (10)      ; restore s; ..
      hv ra13     , NQB          ; if cat then begin
      lk e0       , vk (rd4)    ; fetch first catalog track;
      pp 5        , pm pe0      ;
      pp p1       , LB          ; p:=rel address of date;
      pm pe2      , gm pe0      ; date:=saved date;
      pm ple2     , gm ple0     ; run no:=saved run no;
      pm 1...     , DX          ;
a12: ar e0-1     , t 1          ;
      ar 2        , D LA        ;
      ar 1        , D LB        ; form new checksum;
      sc 39e0     , V LC        ;
      hv ra12     ,            ;
      sk e0       , vk 960      ; write track end;
                                   ....
a13: arn 35      , ca 37        ; if no summation then
d5:  ps [c45-c41], hh -9        ; goto SET TYPEWRITER;
      vk 1        , lk 40       ; fetch track 1;
      pi (rb19)   , ud rb19     ; restore indicator; restore track;
      qq (17)     , t 1         ; prepare track 1 call of SUM; ....
      arnrb19     , tk 10       ; if catalog track was selected then
d6:  ca (rd4)     , hv [c53]    ; goto read and check Help;
      hv 39       ,            ; goto end of track 0;

c1:  qq f        ,            ; checksum word;

e
      ,            ; end special binin;

```

```

b a20
d e7=552 ; binout buf

a0: pp e7 , pmm p ; OUTPUT:
    ar 1.1 D LA ; for i:=words step 1 until 1 do
    ar 1.2 D LB ; begin
b20: qq [sum2]Xt 6 ; sum2:=sum2+6;
a2: cl 32 , ga r1 ; R:=outbuf[p];
[1] sy +1 , it -100 ; M:=marks(outbuf[p]);
b21: bt 0 , hv ra1 ; for j:=1 step 1 until 6 do
    cl -7 , hh ra2 ; writechar(char(j)of:(RM));
a1: pa rb21 , pp p1 ; p:=p+1;
b22: can[words] t -1 ; end;
    hr s1 ; return;
a3: hh ra0 ;

[Bootstrap program, 6 7-bit characters/word. Bits of: Curr instr, prec.]
b23: qq 2.6+ 0.13+ 0.20+ 0.27+ 0.34+ 0.39 ; hv s 25-39, length
    [bits 29-34, 36-38 is length]
    qq 2.6+ 4.13+ 0.20+ 0.27+ 0.34+ 7.39 ; gm s7 t -1 M 25-39, 0-24
    qqf 28.6+ 2.13+ 0.20+ 127.27+ 63.34+ 11.39 ; hv s IKC 19-39, 0-24
    qq 32.6+ 6.13+ 56.20+ 64.27+ 0.34+ 0.39 ; gr xx t 1 MPC 25-39, 0-18
[4b23] qq 0.6+ 0.13+ 0.20+ 0.27+ 64.34+ 10.39 ; pi 0 t -49 31-39, 0-24
    [bits 15-20, 22-24 is 510-length]
    qq 28.6+ 0.13+ 7.20+ 103.27+ 49.34+ 8.39 ; hv xx MR 19-39, 0-30
[6b23] qq 0.6+ 32.13+ 96.20+ 0.27+ 0.34+ 0.39 ; tl 12 V IK 25-39, 0-18
    [bits 22-27, 29-31 is 513-length]
    qqf 24.6+ 44.13+ 1.20+ 32.27+ 64.34+ 2.39 ; bs (r4) IO 19-39, 0-24
    qq 0.6+ 0.13+ 32.20+ 64.27+ 0.34+ 0.39 ; fill , 0-18
    qq 64.20+ 0.27+ 0.34+ 0.39 ; fill

b24: qq 45.39 ; length of track 0
[1b24] qq c1.39-c0.39+1.39 ; length of special binin

c2: vy 32 , sy 14 ; binout track 0 form:
a5: sy 17 , pt -1 ; writetext(<<); sum1:=0;
    pp -1 , pp p1 ;
    pa r1 , it rb23 ; fetch boot;
[1] pm p0 IPC ;
    gm pe7 MPC ;
    bs p503 , hh r-4 ;
b25: pmm rb24 , tl 36 ;
    tk 5 , ac e7 ; pack(length)into:(boot[0]);
    tln 4 , ac e7 ;
    pa ra6 t 507 ;
a6: arn 507 Dt 3 ;
    ck 10 , sr (rb25) ;
    tl -3 , tk 19 ; pack(510-length,boot[4]);
    ac 4e7 , tln 18 ;
    ac 4e7 , ud ra6 ;
    ck 10 , sr(rb25) ;
    tl -3 , tk 12 ;
    ac 6e7 , tln 11 ; pack(513-length,boot[6]);
    ac 6e7 ;
    pa rb22 t 10 ; words:=10;
    pa rb20 t 1 ; sum2:=1;(prepare final 64 mark)
    hs ra0 ; OUTPUT;
    pm(rb25) , gm rb47 ; current length:=length;

```

```

a7:b26:      pmn rc12 t 1      ;OUT BIN0:
              ar 1.1 D LA      ; iword:=iword+1;
              ar 1.2 D LB      ;
              qq(rb20) t 7      ; RM:=(set marks(cell[iword])pos 39+
              ck 3 X            ; marks pos 78) shift 34;
              cl 34 , pp -6     ; sum2:=sum2+7;
a8:          ck -4 , ga r2      ; for p:=-6 step 1 until 0 do
[2]          bs p1 , it 64      ; begin
              sy -1 , pp p1     ; writechar((if p=0 then 64 else 0)+
              bs p511           ; Raddr shift -4);
              cln -6 , hv ra8   ; RM:=RM shift -6
              ; end;

              arn rb47 , sr rb8 ; current length:= current length-1
              sy 64 V LZ        ; if current length=0 then
              gr rb47 , hv ra7   ; goto OUT BIN0;
c17:         it(rb20) , pa -1    ;WRAPUP: R:=sum2x1024+sum1;
              arn -1 , ck 10     ;
              ga r1 , ck -7      ; writechar(bits(3)throu:(9) of:(R));
              sy -1 , ga r1      ; writechar(bits(36,2,R));
              sy -1 , ck -7      ; writechar(bits(29,35,R));
              ga r-1 , sy(r-1)   ;
              vy 17 , hr s1      ; select(t,w); return;

b30:         qq 1.39            ; heading word (initially help)
[1b30]       qq[d37]            ; length
[2b30]       qq 1.39            ; abs base
                                ;
                                ;
c3:          ; procedure binout; begin
b27:         can -1 , hv ra10    ; if no heading yet then
              pa rb26 t c0-b26-1 ; begin binout track 0 form (special binin);
              qq(rb25) t 1      ;
              hs rc2            ;
              pa rb20 t 1      ; sum2:= 1;(prepare final 66 mark)
              vy 32 , sy 13     ; select(p); writechar(13);
              pa rb27 , pt -1   ; no heading yet:= false; sum1:= 0 end;
a10:         gs rb28 , vy 32     ; select(p);
              qq(rb20) t 1      ; sum2:= sum2 + 1;
              pm rb30 IPA      ; disc out:= A mark(heading word);
              sy 1 , pp rb30    ; writechar(1);
              pa rb22 t 1      ; words:=1;
              pa rd7 , hs ra3   ; count:= 0; OUTPUT(heading word);
a11:         hh ra14 LPA        ;NEXT TRACK: if -,disc out then
              pm r2b30 , hs rc8 ; begin select track(abs base);
              arn rb8 , ac r2b30 ; abs base:=abs base+1;
              lk e7 , vk 960    ; read track to(track buf);
              arn r1b30 , sr rb8 ; length:= length-1
              gr r1b30 IZB      ; LZB:= length=0;
a13:         pa rb22 t 40       ;OUT TRACK:
              qq(rb20) t 1      ; words:=40; sum2:= sum2+1;
              sy 40 , hs ra0    ; writechar(40); OUTPUT(trackbuf);
              hv ra11 NZB      ; if -,LZB then goto NEXT TRACK;
b28:         ps[s] , vy 17      ; select(t,w);
              [hr s1 ,]         ; return end;

```

```

a14:  hr s1      , arn rb10 ; else begin bufad:=bufad+40;
a15:  ar rb11    , il      ; il(0, trackbuf, bufad); LZB:=false;
d7:   bt[count] t -1 IZB ; count:=count-1; if count>0 then
      gr rb10    , hv ra13 ; goto OUT TRACK;
      is(rd3)    , it s1   ; count:=block:40;
      pa rd7     ;
      pa rb31    , pm r1b10; rep:=0; bufad:= -40;
a16:  arn rb30   , ga rb29 ;repeat input: unit:= part1(heading word);
      arn r2b30  , tk 28   ;
      ck -10     , ar rd1   ; il(unit,0,blockpart(abs base));
b29:  il[unit]   , arn r1b15;
      is(rb29)   , il s16   ; if status(unit)<0 then
      il         , arn rb15 ; begin rep:= rep+1;
b31:  bt[rep]    Vt -150 LT ; goto if rep>3 then ERROR
      sm rb8     , hh r2    ; else repeat input
      pp rb33    , hh rc7   ; end;
[+2]  hv ra16    , sc r2b30 ; abs base:= abs base + 1;
      ar r1b30   , gr r1b30 ; length:= length - 1; if length<0 then return;
      hv ra15    X NT      ; goto a15
      hv rb28    ; end

```

e

b a20

d e3=-86[search base],e4=40e7[cat in],e5=40e4[cat out],e6=40e5

a:

```

c4:   pm p      , arn ra1 ; procedure writetext(p); integer p; begin
a2:   cl -6     , tk -4   ;next word: M:= cell[p]; R(37:39):= 3 1;
      ga ra3    , ca 15   ;next char: char:= M^63; RM:= RM:64;
      pp p1     , hv ra   ; if char=15 then begin p:= p+1;
a1:   ca 10     , hv s1   ; goto next word end;
b32:  qq[pos]   t 1      ; if char=10 then begin
a3:   qq[char]  , ca 63   ; pos:= pos+1; R:= sum1; writechar(if char=63
      qq(ra3)   t 1      ; then 64 else char); sum1:= R;
      arn -1    , sy(ra3) ; go to next char
      gr -1     , hv ra2  ; end end;

a4:   ; Boolean procedure query; begin M:= sum1;
c5:   pm -1     , lyn ra5 ;L: i:= lyn; if i=<n> then
      ca 37     , hv r4   ; begin writechar(<o>); query:= false
      nc 24     , hv ra4  ; end else if i<y> then goto L else
      sy 53     , sy 18   ; begin writechar(<e>); writechar(<s>);
      gm -1     , hv s1   ; query:= true end; sum1:= M end;
      sy 38     , gm -1   ;
a5:   qq       , hh s    ;

c6:   hs 29e3   ; integer procedure get;
      hh s      X LZ     ; if get word=0 then get:= M else catsum:
c7:   pp rb34   , hs rc4 ; begin writetext(⟨catalog⟩); stop end;
      zq       , hv r    ;ERROR: writetext(⟨fault⟩); stop;

```

b33: k63,29
tfault;

62.

b34: k63,29
tcatalog;

62.

b35: k63,29
tfull;

62.

```

c8:   dln rb7   , ar rb7   ; procedure select track(M); integer M;
      ck -10   , ga r1    ; begin
[1]   vk[group] , cln -20  ; vk(M:960 +960);
b48:  gt r      , vk[track] ; vk(M mod 960)
      hr s1     ; end;

c9:
b36:  gr e5-1   t 1   MPC ;out: oword:= oword+1; cell[oword]:= R;
      it(rb36)  , bs 38e5 ; marks[oword]:= PC; if oword<rel 39 then
      hr s1     ; return;
      grn 39e5   MC      ; cell[rel 39]:= 0; marks[rel 39]:= C;
b49:  pm[reltr] DXt 1     ; reltr:= reltr+1; j:= rel 0 -1;
      pa rb36    t e5-1   ; for sum:= reltr, sum+setmarks(cell[j])
      pa r1      t e5-1   ; while -,LC do j:= j+1;
a6:   ar -1      t 1      ;
      ar 2       D       LA ;
      ar 1       D       LB ;
      sc 39e5    V       LC ; cell[rel 39]:= -sum;
      pm rb38    , hv ra6 ;
      hs rc8     ; select track(work base 1);
      sk e5      , vk 960 ; write track from(rel 0); wait for track;
      arn rb8    , ac rb38 ; work base 1:= work base 1+1;
      sc rb37    ITB      ; work length:= work length-1;
      hv s1      NTB      ; if work length>0 then return;
      pp rb35    , hh rc7 ; writetext(<full>); stop;

e
b a
d a=i
; the following locations overwrite
; the program initialization
b37: qq[work length]
b38: qq[work base 1]
[1b38] qq[work base 2]
[2b38] qq[first free]
[3b38] qq[old length]
b47: qq[current length] ; used in out bin 0
d i=a
e

```

```

b a30
c15:  arm e3      , ga rd0      ;START PROGRAM: ;comment
      gt rd6      , tk 20      ; fetch and spread system parameters:
      ga rd5      , tk 10      ; d18+d17, c53, c45-c41, d53
      ac rd1      , ac r1d1     ; d37 is fetched at 2a1
      ck 10       , ac r1d1     ; d53:40-1
      xr          , dln rb11    ;
      sr rb8      , ck -10     ;
      ga rd2      , ga rd3     ;
      arm 24e3    , ck -10     ; d21;
      ga rd4      , vyn 17     ; select(t,w);
a0:   ar r1c0     t 1          ;
      ar 2        D          LA ; form checksum for special binin;
      hv ra0      NB          ;
      sc rc1      M          ;
c10:                                     ;TRACK 0:
      pp rb39     , hs rc4     ; writetext(<track 0>);
      hs rc5      , hv ra1     ; if querry then begin
      vk 960      , vk 0       ; fetch track 0;
      lk rc11     , vk 960     ; OUT BIN 0(track 0)
b42:  qq e4       , hs rc2     ; outspace(100)
      vy 32       , pa r1     ; end;
      bt 0        t -5        ;
      vy 17       V          ;
      sy 0        , hv r-2    ;
                                     ;HELP:
a1:   pp rb40     , hs rc4     ; writetext(<help>);
      hs rc5      , hv ra2     ; if querry then
      arm 1e3     , tl -30     ; binout(1) length: (d37);
      gr r1b30    , hs rc3     ;
a2:   arm rb42    , ga 14e3    ; assign buffer area for search;
      hs 5e3      IZA         ; fetch first catalog track;
      hv rc7      NZ         ; if R=0 then goto catsum;
a3:   pm(29e3)    X          IPC ;TAKE LAST ITEM: R:= set PC(store[iword]);
      hv ra21     LZ         ; comment see also Help 3 page 13;
      ga rb43     ,          ; if R=0 then goto MAYBE END;
b43:  pi[kind]    t 560       ; in:= kind;
      hh ra6      NTB        ; if special then begin
      hh ra4      NQA        ; if free then begin
a5:   tl -32      , tln 16    ;set work:
      gr rb37     , tln 16    ; work length:= blocks(R);
      gr rb38     , gr r1b38  ; work base1:= work base2:= first block(R);
      arm(29e3)   , hs rc9    ; out(store[iword]);
      hv ra7      NQA        ; if -,free then goto name comes;
      hs rc6      , mb rb44   ; first free:= get^24 0 16 m;
      gr r2b38    , arm(29e3) ; R:= store[iword]^8 m 16 0 16 m;
      ck 8        , tk 16    ; comment booked:= 0;
      ck -24      , hs rc9    ; out(R); goto name comes end free;
a4:   hv ra7      , hs rc9    ;work: out(R);
      hs rc6      , qq        ; R:= get;
      hv ra5      LPA        ; if LPA then secondary work: goto set work;
      [hh ra8     , ]        ; goto next out end special;

```



```

a6:   hh ra8      , t1 -7      ; if kind=drum
      nc 0        , ca 1      ;   Vkind=disc then
      pp e6-1     , hh ra9     ;   goto STACK ITEM;
      t1 7        , hs rc9     ;   out(R); R:= get;
a8:   hs rc6      , hs rc9     ;next out: out(R);
a7:   hs rc6      , qq        ;name comes: R:= get;
[1a7] qq[see 1a10] V LZ      ; goto if R#0/NA then
      hh ra8      NA          ; next out else TAKE LAST ITEM;
a9:   hv ra3      , gp rb46    ;STACK ITEM:=
      t1 7        , ud ra10    ; istack:= stack bottom;
      hs rc6      , qq        ; for R:= R, get, get while NA/NA do
a10:  ; begin
b46:  gr[istack] t 1 MPC      ; store[istack] MPC:= R;
      hs rc6      , ud r1a7    ; istack:= istack+1
      hv ra10     NA          ; end;

      pp 1e6      , pm p      ; pos:= 0; M:= sum1; writecr;
      pp p1       , LA        ; sum1:= M; in:= bits and kind;
      pm -1       , sy 64     ;
      gm -1       , pi(rb43)  ; writetext(address(stack[if Amark(
a11:  pa rb32     , hs rc4     ;   stack[1]) then 2 else 1]));
      bs(rb32)    t 7         ; for pos:= pos+1 while pos<7 do
      sy 0        , hv ra12    ; writechar(0);
      qq(rb32)    t 1         ; writechar(0);
      sy 0        , hv ra11    ;
a12:  hs rc5      , hv ra17    ; if -,query then goto NO;
      pm e6       X          ; YES: R:= stack[1]; comment the areaword;
      hv ra13     LPB        ; if -,reserved then begin
      t1 -32      , tln 16    ; length:= blocks(R);
      gr r1b30    , tln 4     ; heading word:= first block(R)+
      t1 12       V NTA      ; (if disc area then unit(R) else 0) pos 9+
      ck 18       , t1 12    ; (if disc area then 1 else 0) pos 40;
      gr rb30     MTA        ;
a16:  mb rb44     , gr r2b30   ; abs base:= first block(R);
      ps ra19     , hv rc3    ; binout; goto OUT STACK
      ; end else begin
a13:  pm r2b38    , mb rb44    ; heading word:=
      gm rb30     MTA        ; if disc area then
      gr r2b30    , tk 18     ;
      hv ra14     LTA        ; first free-unit(first free)
      t1 23       ,          ; +unit(stack[1]) pos 9+1 pos 40
      arn e6      , ck -16    ; else first free;
      t1 16       , hv ra15   ; abs base:= stack[1]^24 0 16 m;
a14:  ga rb30     ,          ; stack[1]:= stack[1]^24 m+first free;
      t1 27       , arn e6    ;
      ck -12      , t1 12     ; length:= blocks(stack[1]);
a15:  gr e6       , ck 8      ; first free:= first free+length;
      t1 -24      , gr r1b30  ;
      ac r2b38    , hv ra16   ; binout; goto OUT STACK end YES;

```

```

a17:  hv ra3          LPB ;NO: if reserved then goto TAKE LAST ITEM;
d a19=i-1              ;OUT STACK:
a18:  pa rb45      t e6-1 ; for j:= stack bottom
b45:  pm[j]        Xt 1   IPC ; step 1 until istack do
      hs rc9              ; out(store[j]);
      pm rb45      X      ;
      nc(rb46)     , hv rb45 ;
      hv ra3              ; goto TAKE LAST ITEM;

a21:  pa r2         Vt 39 LB ;MAYBE END: if +;LB then
      hs rc6        , hv ra3 ; begin get; goto TAKE LAST ITEM end;
                        ;END SCAN:

[-1]  bt 39         t -1    ; for j:= 1 step 1 until 39 do
      ps r-2        , hvn rc9 ; out(0);
      pm r1b38      , hs rc8  ; select track(work base2);
      lk e4         , ud rb48 ; read track;
      arn e4        , ck 2    ; R:= free areaword;
      pm e4         X      IOB ;
      ck 8          , tl -24  ; old length:= blocks(R);
      gr r3b38      , XV     NOB ; free areaword:=
      tln -23       , hh ra22 ; (if free kind=drum then
      tln 4         , sr r2b38 ; (first block(R)+old length
      ar r3b38      , ar r2b38 ; -first free)pos23+first free
      tk 16         , ar r2b38 ; else
a22:  hv ra23      , arn r2b38 ; (first block(R)+old length
      sc r3b38      X      ; -(first free^28 0 12 m))pos23+first free
      tl -27        , ar r3b38 ; +free kind pos2)
      tk 16         , ac r2b38 ; +special+booked to be set;
      pm e4         , tl 23   ;
      arn r2b38     , ck -16   ;
      tl 4          , ck 12   ;
      it 1.2        ;
a23:  ar 1.3+1.6 D  ;
      gr e4         , it 1    ; reltrack in search:= 1;
      pa 15e3       , hs 15e3 ; sum track;
      ud rb48       ;
      sk e4         , vk 960  ; write track; wait for track;
      pm 24e3       , arn rb49 ; heading word:= d21 pos39+1 pos41;
      gm rb30       MB      ; abs base:= work base 2;
      pm r1b38      , gm r2b30 ; length:= reltr pos 39;
      tl -30        , gr r1b30 ;
      pp rb41       , hs rc4   ; writetext(<catalog>);
      hs rc5        , hv r2    ; if query then
      hs rc3        ; binout;
      vy 32         , sy 66    ; select(p); writechar(66);
      hs rc17       ; WRAPUP;

      pa rb25      t b24-b25 ;PREPARE NEXT RUN:
      pa rb26      t c13     ; prepare track 0 output;
      pm r1b10     , gm rb10  ;
      grnrb15      M        ;
      pm rb8       , pa rb49  ;
      pa rb36      t e5-1    ; prepare out;
      gm rb30      M        ; prepare help output;
      gm r2b30     , it -1    ; no heading yet:= true;
      pa rb27      , hv rc10  ; goto TRACK 0;

```

b44: qq 1.23-1.39 ; mask 24 0 16 m;

b39: k58,62,63

t

track 0 ;

b40: t

help t ;

b41: t

catalog ;

d c12=i-1,c13=c12-b26

vk 960 , vk 0 ; track 0 heading

ly r , ly r ;

ly r , sk rc11 ;

lk 0 , vk 0 ;

hv 1 ;

c11:

e ; end program block

c16: hs 1 ;

hsf 2 ;

tbinout; ;

qq 16 , ; binout, 0 10..c11<

qqf 10.19+c11.39 ;

qqf , ;

[STOP,SUM]

isystem punch

ec16

```
;slip<
[29.7.67
[STOP, CLEAR]
b a15, b15
i=10
```

(2)

Check bin, page 1]

```
a4: vy 17 , pa rb2 ; START: select(t, w); sum:= 0;
    pp rb , hs ra6 ; message({<check bin});
b1: pa[chars] D ; chars:= 0;
    ly ra , vy 16 ; typechar; select(r, w);
a: lyn DV ; search begin: read symbol;
    pp rb3 , hv ra5 ; if parity error then alarm({<parity});
    ca 13 , hv ra1 ; if aa then goto binin form;
a2: nc 14 , hv ra ;
    lyn ra V ; if not < then goto search begin;
    pp rb3 , hv ra5 ;
    nc 17 , hv ra2 ;

    pp 1 , hsn ra7 ; BIN 0 FORM:
    pp 1 ;
    tk 1 , hs ra7 ;
    tk 3 , ga rb4 ; words:= read(1), read(1);
    pp 59 , hs ra7 ; read(59); skips bootstrap + 1 char;
    pp 7 ; for i:= words step -1 until 1 do
b4: bt[words] t -1 ;
    ps r-3 , hv ra7 ; read(7);
    hv ra8 ; goto check sum;

a1:
a1h: pp 1 , hs ra7 ; BININ FORM: read head symbol;
    ga rb4 , it 63 ; if symbol < 63 then
    bs(rb4) , hv ra3 ; begin
    pp 6 ; for i:= symbol step -1 until 1 do
    bt (rb4) t -1 ; read(6);
    ps r-3 , hv ra7 ; goto binin form;
    hv ra1 ; end;
a3: ca 66 , hv ra8 ; if symbol = end then goto check sum;
    nc 64 , ca 65 ; if symbol = repeat V symbol = label then
    pp 4 , hh ra1 ; begin read(3); goto binin form end;
    pp rb5 , hv ra5 ; alarm({<tape sum});

a7: it p , qq (rb1) ; procedure read(p); begin chars:= chars + p;
a9: tl -7 , pp p-1 ; for p:= p step -1 until 1 do begin
    ly ra V ; RM:= (RM shift -7) + lyn;
    pp rb3 , hv ra5 ; if parity error then alarm({<parity});
b2: ac [sum] D ; sum:= sum + char;
    bs p , hv ra9 ; end;
    hr s1 ; end read;

a8: it (rb2) , pt rb6 ; CHECK SUM: saved sum:= sum;
    pp 3 , hs ra9 ; read(3);
    tk 4 , ca (rb1) ; if chars ≠ first part
b6h: ck 10 , nc -1 ; V saved sum ≠ second part then
    pp rb5 , hv ra5 ; alarm({<tape sum});
    hv ra4 ; goto start;
```

```

a5: sy 29 , ps ra4-1 ; ALARM: writered; prepare return to start;
a6: sy 64 , sy 58 ; MESSAGE: writecr;
a11: pmm p X ; p points to text;
a10h:cl 34 , ck -4 ; rep:
      ga rb7 , ca 15 ; Raddr:= next char;
      pp p1 , hv ra11 ; if Raddr = 15 then get next word;
      ca 10 ; if Raddr = end text then
      sy 62 , hr s1 ; begin writeblock; return end;
b7: sy -1 , cl -6 ; write char;
      hh ra10 ; goto rep;

```

```

b: tcheck bin; ;
b3: tparity; ;
b5: ttape sum; ;

```

```

a12: hs 1 ; ENTRY AFTER SLIP LOADING:
      hsf 2 ;
      tbinout; ; binout, 0 first ..top<
      qq 16, ;
      qqf a4.19+a12.39 ;
      qqf, . ;

```

```

[STOP, SUM] k i check bin
e a12

```

```
;slip<
[4.7.67
[STOP, CLEAR]
```

Cattap, page 1]

```

b a11 ;
i=483, a6=512 ; check last in cell 512
a11: vy 17 sy 64 ; START:
      sy 51 sy 49 ; writetext(<<cattap>>);
      sy 19 sy 19 ;
      sy 49 sy 39 ;
      ly a' pp (a) ; p:= station:= typechar;
a: ly -1 D ; if typechar # comma then
      nc 27 hv a11 ; goto start;
      arn a1 us 0 ; buffer[1]:= <<cattap>>;
      usn p64 arn a2 ; rewind;
      us p iln p160 ; writelabel;
      arn a3 il 0 ;
      arn a4 ; if error then
      hv a5 NT ; begin
a7: sy 64 sy 54 ; write error:
      sy 49 sy 20 ; writetext(<<fault>>);
      sy 35 sy 19 ; goto start
      hv a11 ; end;
a5: arn a6 us 0 ; buffer[1]:= EOF;
      arn a2 us p144 ; write EOF even parity;
      iln p160 arn a3 ;
      il 0 arn a4 ;
      hv a11 NT ; if error then goto write error;
      hv a7 ; goto start;

a8: tcattap; ;
a1: qq a8.9+1.19+1.39 ;
a9: qq 15.5+15.11+15.17+15.23 ;
a2: qq 1.19+1.39 ;
a4: qq ;
a3: qq a4.9+1.19+0.39 ;
a6: qq a9.9+1.19+1.39 ;

a10: hs 1 ; ENTRY AFTER SLIP LOADING:
      hsf 2 ;
      tbinout; ; binout, 0 first..last<
      qq 16; ;
      qqf a11.19+a6.39 ;
      qqf. ;

[STOP, SUM] a i cattap
e a10
```

;30.6.67. Punch head bin0 PM.

```
bi=10,a5
  arn a3      t-1      ;
  tk 3        ,gr (i-1) ;skift de første 4 ord som indlæses med kanal 0
  bt 3        t-1      ;3 skift til venstre.
  vy 32       ,hv i-3   ;
  a2: pmm a    IRC t-1   ;og vælg strimmeloutput
  ar 256      D LRA      ;et ord til M
  ar 128      D LRB      ;Amerke
  xr          ,tl -10    ;og Bmerke i R
  ar 128      D          ;
  a4: qq 440   t-1      ;sæt ord slut mærke
  tl -7       X IZA      ;tæl karakterer
  gt 1        ,sy        ;næste karakter i tæledel
  hv (a4)     XVD NZA t-1;hul næse karakter
  bs (a2)     ta1        ;hop med karaktertælling hvis flere karakterer i dette ord
  hv a2       ,sy 49     ;hvis flere ord
  bt (a4)     t-2        ;hop til a2
  sy 6        ,hh i-2    ;fyld op med mellemslag til ca 435 karakterer
  ly a5       ;
  nc 14       ,hv i-1    ;læs en karakter
  a5: sy       ,ly 1     ;skip indtil understregning
  nc 11       ,hv a5     ;kopier
  sy 11       ;indtil stopcode
  bt 100      t-1        ;hul stopcode
  sy         ,hv r-1     ;
  hsf 2       ,hv r-1    ;hop ca 100 mellemslag
                        ;hop ud til help

a1: ly        D          ;det følgende læses med gammel kanal 0
  nc 11       ,hv r-1    ;skip indtil stopcode
  qqn         ;
  tl -6       ,ca 0      ;primitivt læseprogram
  ly r4       ,hs r-1    ;
  gm s3       M t-1      ;

  tl 3        ;boot
  gr r-a1+2 MRC t-1      ;
  hv 34       ;
  gr 42       MRC t-1    ;
a3: 272/351/ 68/ 3a     ;
a: e10          ;
```

det følgende kopieres til strimmel

```
<42      2m
i=10,vyn16
tl-6,ca
lyr4,hsr-1
gms3t-1M
e10
```

```

;slip<
[Punch head kompud. PM 26.9.67 version 2
SET TAB | |clear code ]

```

```

b b16,d,e88
d=1
i[redefine
;sum code
a;clear code ]
d=d+d+d+d+d, d=d+d+d+d, d=d+d+d+d, d=d+d+d+d, c38=d-320; d=antal kanaler

b a7
i=10, vy 32 ,pp ;punch boot
sy 14 ,sy 17 ;≤
a2: pmn b X t-1 ;næste ord
bs (a2) tb16 ;
cl 1 ;
a1: cl 36 ,tk -4 ;næste char i R0-9
bs pb1 tb2 ;
qa V ;
bs pb3 tb4 ;
ar 64 D ;sæt ordmærke
ga a ,cl -6 ;
a: sy ,bs pb5 ;hul char
pp pb6 ,hh a1 ;tæl char
bs (a2) tb8 ;
pp pb10 ,hv a2 ;tæl ord

a3: qa 6 ,pt -1 ;punch kompind
a4: pmn b8 IRC t-1 ;næste ord
ar 256 D LRA ;A-mærke
ar 128 D IRB ;B-mærke
xr ,tl -10 ;
ar 128 D ;
tl -7 X IZA ;
gt r ,sy ;
hv r-2 X NZA ;
bs (a4) tb11 ;
hv (a3) VD t6 ;
it (a3) ,pa -1 ;
arn -1 ,gt a5 ;checksøm og antal
a5: tk -7 ,sy ;
a6: gt a6 ,sy ;
tk 3 ,ga a7 ;
a7: sy ;
bt 100 t-1 ;mellemslag
sy ,hv r-1 ;
vy 17 ,ly r3 ;vent
nc 51 ,hsf 2 ;
vy 32 ,ly r1 ;stop ,kopier
sy ,hh r-1 ;
e ;hullesekvens slut

b a23, c6
b11: lyn ra9 ,tk -7 ;kompind som hulles
ly ra9 ,tk -7 ;læs checksum og antal for kompind
ly ra9 ,tk 4 ;
gs ra21 ,ca (sb15) ;
tk 10 ,nc (sb12) ;
hv ra21 ;sumfejl i kompind
ly ra9 ;skip til <
nc 17 ,hv r-1 ;
e: pa a23 t51 ;
ly ra9 ;
nc 62 ,hv ra20 ;hvis ikke 62 hop til læsning
[

```



```

b a3
[-2]   arm ra      IRC t1
      gr 33      MRA t1
      hv r-2     NRB
      arm ra1    ,pi 2
      gr 28      MA
      arm ra3    ,gr 9
      pmm ra2
a:     hh 35      X
[34]   pmm 1.3    XD IZA
[35]   t1 -7     ,ly r1
a3:
[36]   pi        IZA t508
[37]   xr        X IZB
[38]   hv 35     IZB
[39]   grf 41    MRC t-1
a1:    arm 999   ,hv 995
a2:    qq 62.16+17.23+14.37+1.38
e
e38=200,e39=240,a4=280
a20:   pa ra19   ,pa ra13
a9:    pi        ,hs ra
a8:    pm ra5    ,ga ra3
      ga rc6    ,tk 10
      ga ra2    ,tk 10
      gr a4     X MRA
      hv ra6    LRB
a23:   bt 51     V NQB t-1
c6:    pp p[ac]  ,hv ra18
      ly ra9    ,hv r-2
e44:   nt 38     ,qq (ra2)
a18h:  pp (ra3)  ,it -2
      pi 1      ,bs p-39
      pp p-40   ,it 1
      can (ra2) ,hv ra10
      it (re44) ,vk (ra2)
c4:    lk e38    ,arm ra2
      hs rc1     ,hv ra12

c:     gs r4c    ,ps (re44)
[1c]   bs s-831e88 t-832e88
      vk s1     ,lk (rc3)
      pt r1c    t511
[4c]   ps       ,hh s

a5:    hsf 2
a7:    hs ra     NB
a14:   gr se39   MRC
      hk rc     ,pp p1
      bs p472   ,hv rc2
      hs rc     ,pp
a6h:   hs rc1    ,it -1
a2:    btf [ak]  ,it 40
c2h:   qq (ra3)  ,ps p
a3:    bt [ac]   IRCT-1
a12h:  hv ra7    ,ga re44
a10:   hv ra20   NQB
      arm a4    IRA
      hh ra8    NZ
      hv ra9    NRA

```

;ellers dan gammel kanal 0 læseprogram

;

;

;

;

;

;

;

;

;

;resten af blokken er konstanter

;

;

;

;

;

;

;

;

;

;de tre allerede læste karakterer

;

;definer kanal buffere

;sum og antal=0

;læs etikette

;hop ikke overspringelse

;hvis startetikette tæl melleslag

;ellers tæl i p og hop

;hop hvis samm kanal

```

        lyn ra16      ,tk -7      ;læs sum og antal
        ly ra16       ,tk-7
        ly ra16       ,tk4
        nc (ra19)     ,hv ra21
        tk 10         ,ca (ra13)
        hs rc1        ,hv ra17

a21:    ps           ,vy 17      ;fejlreaktion
        sy 29         ,sy 64
        sy 1t[s]      ,sy 20[u]
        sy 36[m]      ,arn r3
        gr sb13       ,ly r
        vy 32         ,hvn sb13
        gm s3         M t-1

c1:     pt r1c       IRB t-832e88 ;start skrivning paa kanal
        it (rc3)      ,pa ra14
c3:     nt e38       ,qq e38+e39
        vk (re44)     ,it 1
        is (re44)     ,it s+511
        bs 832e88     ,sk (rc3)
        hh s

a:      pmn 1,3      XD IZA      ;læs et ord
        tl -7        ,ly ra16
a16:    pi          IZAt-516
a13:    ac          DX
        hv r1a       XLZ
a19:    xr          t6
        tl 3         ,hr s1

a17:
e

b a12                                ;læs udhopsadresse mm

a:      lyn         D           ;
        nc 14       ,hv ra      ;skip til _
        lyn ra      ,ca 35      ;l
        vy 16       ,hv ra      ;
        ca 17       ,hv e       ;≤
        ca 20       ,hv ra8     ;u
        ca 53       ,hv ra9     ;e
        vy 17       ,sy 64      ;e
        sy 58       ,hv ra      ;

a8:     hsn ra10 X      ;u læs adresse
        hv ra        ;

a10:    it ra6       ,pt ra4     ;adresselæsning
a1:     lyn ra2      ,ga ra2     ;
        ca          ,hv ra1     ;mellemslag
        ca 32       ,sc ra12    ;-
        ca 16       ,hvn ra7    ;nul
a4:     ca 64       ,hv         ;vr
a2:     bs          t9          ;
        hv ra1      ;ikke ciffer
a7:     it ra3       ,pt ra4     ;ciffer
        mt ra12     ,ml ra5     ;
        tl 39       X NZ       ;
        hv ra1      ;
a3:     xr          ,ga ra11    ;gem adresse
a6:     hv s1        ;

a5:     10          ;
a12:    qq          ;
[

```

```

b a2 ;
a9: hsn ra10 X ;e læs adresse
b14: vk 960 ,vk 319e88;hent ferritlagerbillede
a: lk 1000 ,it 1 ;
vk 293e88 ,it 40 ;
lk -40 ,it -1 ;
bt 23 ,hh ra ;
a1: arn r a2 t1 IRC ;sæt celle 6 til 9
gr 5 t1 MRC ;
bt 3 t-1 ;
hv ra1 ;
a2: vk 318e88 ,hv 6 ;

```

```

lk 960 ,vk ;
qq ;
vy 17 ;
a11: qq ,hv (r) ;
e ;
e ;
b a2 ;

```

```

b8: pa ra t1.3 ;
a: pmm 1.3 XD IZA ;
a2: t1 -7 ;
ly ra1 ;
a1: pi LZA t-516 ;
ac XD ;
hv ra2 X LZ ;
xr t6 ;
t1 3 ;
gr 960b8-b14MRC t-1 ;
bt b8-1b11t-1 ;
hv ra ;
b16: hs 960b11-b14 ;
b7: t1 10 ;
gr s5b7-b8M t-1 ;
hv s ;
gm s4 M t-1 ;
hv s ;

```

```

b: b12=2a1-b7, b13=3b8-b7, b15=4a1-b7 ;
b6=b-b8, b9=b6+b6+b6+b6, b2=511b7-b, b1=b2-b9 ;
b9=b9+b6+b6, b4=511b8-b7, b5=-512b9, b3=b5-b6, b10=-b9+1 ;
e ;
e10 ;SUM CODE 1

```

[4.7.67

Create new -> old, page 1]

b a10, b10

i=10

```

vk 960 , vk 0 ; START: read track 0;
lk b , vy 33 ; select(p, t);
pa b2 t 1 ; chars:= 1;
ly D ; typechar;
sy 14 , sy 17 ; punch head;
pt -1 , vk 0 ; sum:= 0;
a2h: pp 0 , pmn pb10 ; for p:= 0 step 1 until 9 do
ar 1.1 D LA ; begin
ar 1.2 D LB ;
b2: qq 1 Xt 6 ; chars:= chars + 6;
a1h: cl 32 , ga r1 ; R:= boot[p]; M:= marks.2;
sy -1 , it -100 ;
bt 0 , hv r2 ; punch the 6 characters;
cl -7 , hh a1 ;
pa r-2 , pp p1 ;
ncn p-10 , hh a2 ; end;

a3h: pp 0 , pmn pb1 ; for p:= 0 step 1 until 41 do
ar 1.1 D LA ; begin
ar 1.2 D LB ;
qq (b2) t 7 ; chars:= chars + 6;
ck 3 X ; R:= code[p]; M:= mark.39;
cl 34 , ps -6 ;
a: ck -4 , ga r2 ;
bs s1 , it 64 ; punch the 7 characters,
sy -1 , ps s1 ; the last with a 64-hole;
bs s , hv a4 ;
cln -6 , hv a ;
a4: pp p1 ;
ncn p-42 , hh a3 ; end;

sy 64 , it (b2) ; punch tail mark
pa -1 , arn -1 ;
ck 10 , ga r1 ; and chars, sum
sy -1 , ck -7 ;
ga r1 , ck -7 ;
sy -1 , ga r1 ;
sy -1 , hv 10 ; goto start;

```

[Bootstrap program, 6 7-bit characters / word. Bits of: Curr instr, prec.]

```

b10: qq 2.6+ 0.13+ 0.20+ 0.27+ 5.34+ 4.39 ; hv s 25-39, length
[bits 29-34, 36-38 is length]
qq 2.6+ 4.13+ 0.20+ 0.27+ 0.34+ 7.39 ; gm s7 t-1 M 25-39, 0-24
qqf 28.6+ 2.13+ 0.20+ 127.27+ 63.34+ 11.39 ; hv s IKC 19-39, 0-24
qq 32.6+ 6.13+ 56.20+ 64.27+ 0.34+ 0.39 ; gr xx t1 MPC 25-39, 0-18
[4b10] qq 0.6+ 0.13+ 58.20+ 32.27+ 64.34+ 10.39 ; p1 0 t -49 31-39, 0-24
[bits 15-20, 22-24 is 510 - length]
qq 28.6+ 0.13+ 7.20+ 103.27+ 49.34+ 8.39 ; hv xx MR 19-39, 0-30
[6b10] qq 0.6+ 32.13+ 64.20+ 58.27+ 56.34+ 0.39 ; t1 12 V 25-39, 0-18
[bits 22-27, 29-31 is 513 - length]
qqf 24.6+ 44.13+ 1.20+ 32.27+ 64.34+ 2.39 ; bs (r4) IO 19-39, 0-24
qq 0.6+ 0.13+ 32.20+ 64.27+ 0.34+ 0.39 ; fill , 0-18
qq 64.20+ 0.27+ 0.34+ 0.39 ; fill

```

```

b1: vk 960 , vk 0 ; code: select and write track 0;
sk r1 , hv r-1 ; goto code;
b: e 10 ; track 0 is read to here;

```